Hacettepe University

Department of Electrical and Electronics Engineering

ELE 489: Fundamentals of Machine Learning

HW-2 Report

Kıvanç Ateş

2210357113

Q1)

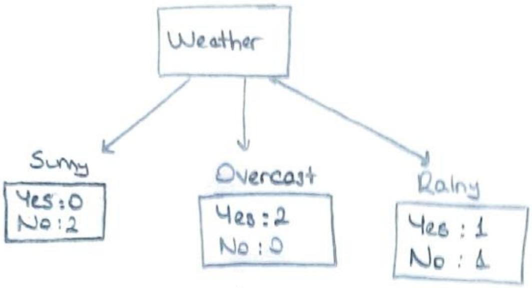
i) $GINI_{total} = 1 - \sum_{i=0}^{1} (p_i(+))^2$

$p_0(+) = \dfrac{\text{Play Outside} \rightarrow \text{No}}{\text{total samples}} = \dfrac{3}{6} = \dfrac{1}{2}$

$p_1(+) = \dfrac{\text{Play Outside} \rightarrow \text{Yes}}{\text{total samples}} = \dfrac{3}{6} = \dfrac{1}{2}$

$GINI_{total} = 1 - (p_0(+))^2 - (p_1(+))^2 = 1 - \frac{1}{4} - \frac{1}{4} = 0.5$

ii) Split the entire dataset by weather:



$GINI_{sunny} = 1 - \sum_{i=0}^{1} (p_i(+))^2$, $p_0(+) = \frac{2}{2}$, $p_1(+) = \frac{0}{2} = 0$

$GINI_{sunny} = 1 - p_0^2(+) - p_1^2(+) = 1 - 1^2 - 0^2 = 0$

$GINI_{overcast} = 1 - \sum_{i=0}^{1} (p_i(+))^2$, $p_0(+) = \frac{0}{2} = 0$, $p_1(+) = \frac{2}{2} = 1$

$GINI_{overcast} = 1 - p_0^2(+) - p_1^2(+) = 1 - 0^2 - 1^2 = 0$

$GINI_{rainy} = 1 - \sum_{i=0}^{1} (p_i(+))^2$; $p_0(+) = \frac{1}{2}$, $p_1(+) = \frac{1}{2}$

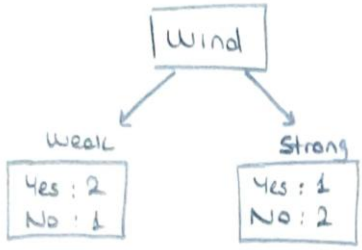
$GINI_{rainy} = 1 - p_0^2(+) - p_1^2(+) = 1 - \frac{1}{4} - \frac{1}{4} = 0.5$

$GINI_{weather} = \sum_{i=1}^{3} \frac{n_i}{n} GINI(i) = \frac{n_1}{n} GINI(1) + \frac{n_2}{n} GINI(2) + \frac{n_3}{n} GINI(3)$

$n_1 = 2, n_2 = 2, n_3 = 3, n = 6$, $GINI(1) = 0$, $GINI(2) = 0$, $GINI(3) = 0.5$

$GINI_{weather} = \frac{2}{6} \times 0 + \frac{2}{6} \times 0 + \frac{2}{6} \times 0.5 = 0.1667$

iii) Split the entire dataset by wind :



$$GINI_{weak} = 1 - \sum_{i=0}^{1} (p_i(t))^2 , \quad p_0(t) = \frac{1}{3} , \quad p_1(t) = \frac{2}{3}$$

$$GINI_{weak} = 1 - p_0^2(t) - p_1^2(t) = 1 - \frac{1}{9} - \frac{4}{9} = 0.4444$$

$$GINI_{strong} = 1 - \sum_{i=0}^{1} (p_i(t))^2 , \quad p_0(t) = \frac{2}{3} , \quad p_1(t) = \frac{1}{3}$$

$$GINI_{strong} = 1 - p_0^2(t) - p_1^2(t) = 1 - \frac{4}{9} - \frac{1}{9} = 0.4444$$

$$GINI_{wind} = \sum_{i=1}^{2} \frac{n_i}{n} GINI(i) = \frac{n_1}{n} GINI(1) + \frac{n_2}{n} GINI(2)$$

$$n_1 = n_2 = 3 , \quad n = 6 , \quad GINI(1) = GINI(2) = 0.4444$$

$$GINI_{wind} = \frac{3}{6} \times 0.4444 + \frac{3}{6} \times 0.4444 = 0.4444$$

iv) $GINI_{weather}$ is less than $GINI_{wind}$. Therefore, Weather should be choosen as the root node.

Q2) Statistical metrics like variance, skewness, kurtosis, and entropy can be used to analyze images and provide information about their complexity, texture, and brightness distribution. Applications like image processing, computer vision, and pattern recognition make extensive use of these capabilities.

Variance

The variance quantifies how widely the intensity values of pixels deviate from the mean. It measures the degree of deviation from uniform intensity in an image. Variance is used in contrast enhancement, noise detection, and edge detection. High variance means the image has high contrast and sharp details. Low variance means the image has low contrast and smooth or uniform regions. The mathematical formula to calculate the variance of image is given below.

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (I_i - \mu)^2$$

I_i is the intensity of pixel i.

μ is the mean intensity.

N is the total number of pixels.

Skewness

Skewness describes the asymmetry of pixel intensity distribution. It is used in brightness balancing and segmentation. Positive skewness means the image contains more dark pixels. Negative skewness means the image contains more bright pixels. Near zero skewness means the image has balanced brightness distribution. The mathematical formula to calculate the skewness of image is given below.

$$S = \frac{1}{N} \sum_{i=1}^{N} (\frac{I_i - \mu}{\sigma})^2$$

S>0 (Right-skewed): More dark pixels, bright regions are outliers.

S<0 (Left-skewed): More bright pixels, dark regions are outliers.

S=0 (Symmetric): Even brightness distribution.

Kurtosis

Kurtosis measures the tailedness of the pixel intensity distribution, illustrating the behavior of extreme values. It is used for edge detection and texture classification. A higher kurtosis indicates a high contrast and sharp edges in the image. A lower kurtosis indicates that the image is smooth and has consistent intensities. The following is the mathematical formula for determining an image's kurtosis.

$$K = \frac{1}{N} \sum_{i=1}^{N} (\frac{I_i - \mu}{\sigma})^4$$

High kurtosis (K > 3): More extreme pixel intensities (sharp peaks).

Low kurtosis (K < 3): More uniform, less extreme intensities.

Entropy

The degree of information or unpredictability in an image is measured by entropy. It is employed in cryptography, texture analysis, and image compression. Images with high entropy are complex and complicated. Simple, smooth images with fewer variations are those with low entropy. The following is the mathematical formula for determining an image's entropy.

$$E = -\sum_{i=0}^{255} p_i \log_2 p_i$$

We have 4 features in this dataset. These are variance, skewness, kurtosis and entropy. The visualization of features in groups of two is shown in Figure-1. The data set is too large and complex to be separated simply and linearly. Decision tree can be used to classify large and complex data sets. When the point distribution graphs of the features are examined as binary, some binary data groups can be used effectively in classifying the data. Because these pairs are separated from each other and do not overlap much. These can be given as examples of these pairs: Variance vs Skewness, Variance vs Kurtosis, Skewness vs Entropy.

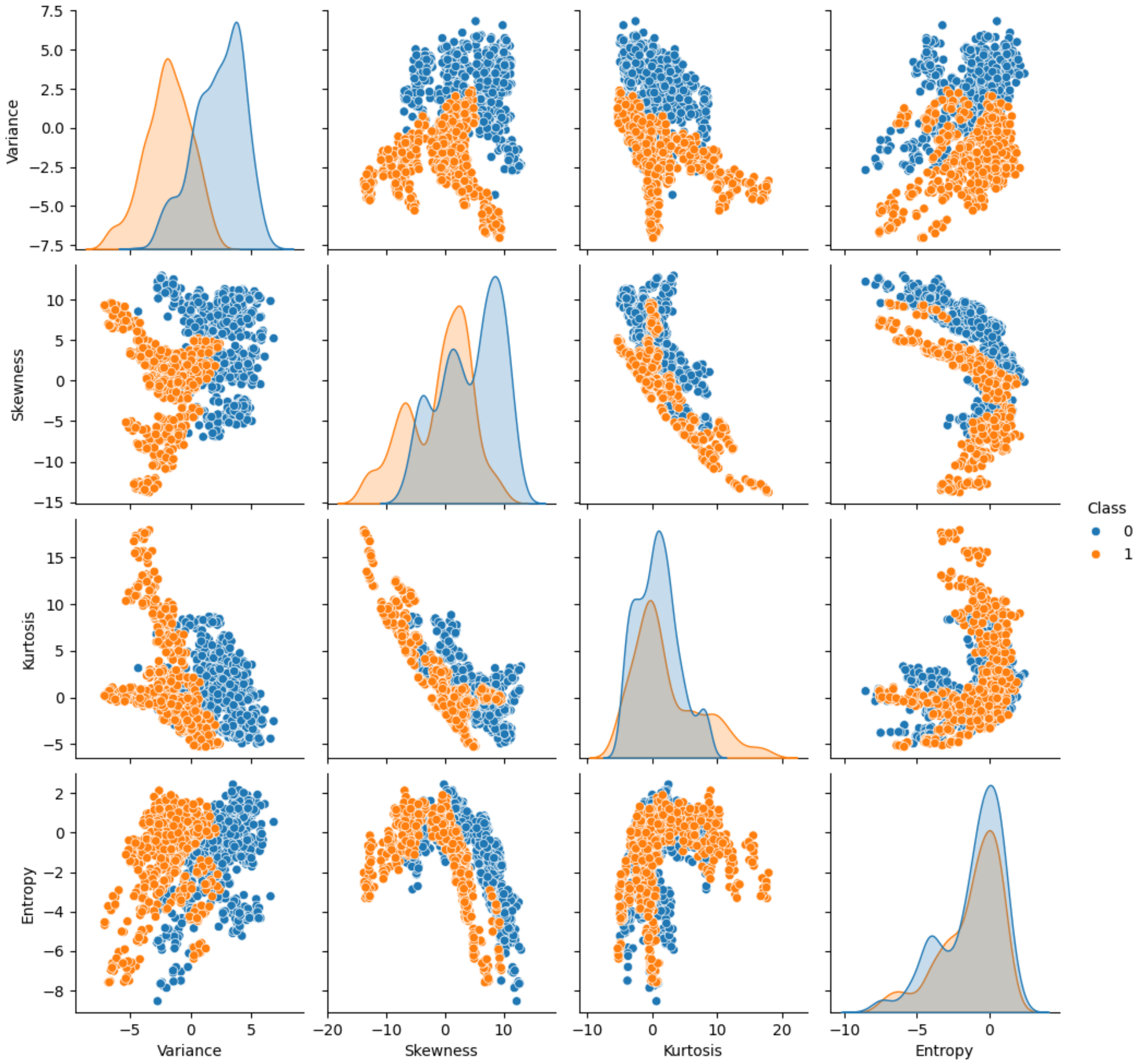


Figure-1

First, I separated 1372 data into 275 test data and 1097 training data. Then, I trained the model using the training data and tested the accuracy of the model using the test data. We need to test the model for different “max_depth”, “min_samples_split” and “criterion” values. First, we need to understand how the model changes when we change these values. “max_depth” parameter specifies the maximum depth of tree. Its default value is “none” means the nodes are expanded until all leaves are pure or until all leaves contain less than “min_samples_split” samples. When “max_depth” has small value (e.g., 2-5), the tree is less complex and there might be underfit. But “max_depth” has large value (e.g., 10+), the tree is deep and captures more patterns. There is risk of overfitting. “min_samples_split” specifies the minimum number of samples required to split an internal node. Its default value is 2. When “min_samples_split” has small value (e.g., 2-5), the tree goes freely. So, the tree has more splits and complex model. There is a higher risk of overfitting. But “min_samples_split” has large value (e.g., 10-20), the tree grows more conservatively. So, the tree has fewer splits and simple model. There is a risk of underfitting. “criterion” parameter measures the quality of split. Its default value is ‘gini’. ‘gini’ measures Gini impurity. It is faster to compute. But makes slightly less pure splits than entropy. ‘entropy’ is slower but might produce slightly better splits.

First, I test and train the tree for default parameters. These are “max_depth = none “, “min_samples_split = 2” and “criterion = ‘gini’”. These values give complex and depth tree, but it is computationally fast because I used ‘gini’ criterion. Classification report, accuracy and confusion matrix are shown in Figure-2 for these values.

```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.99      0.99       157
           1       0.98      1.00      0.99       118

    accuracy                           0.99       275
   macro avg       0.99      0.99      0.99       275
weighted avg       0.99      0.99      0.99       275

Accuracy: 0.9927
```

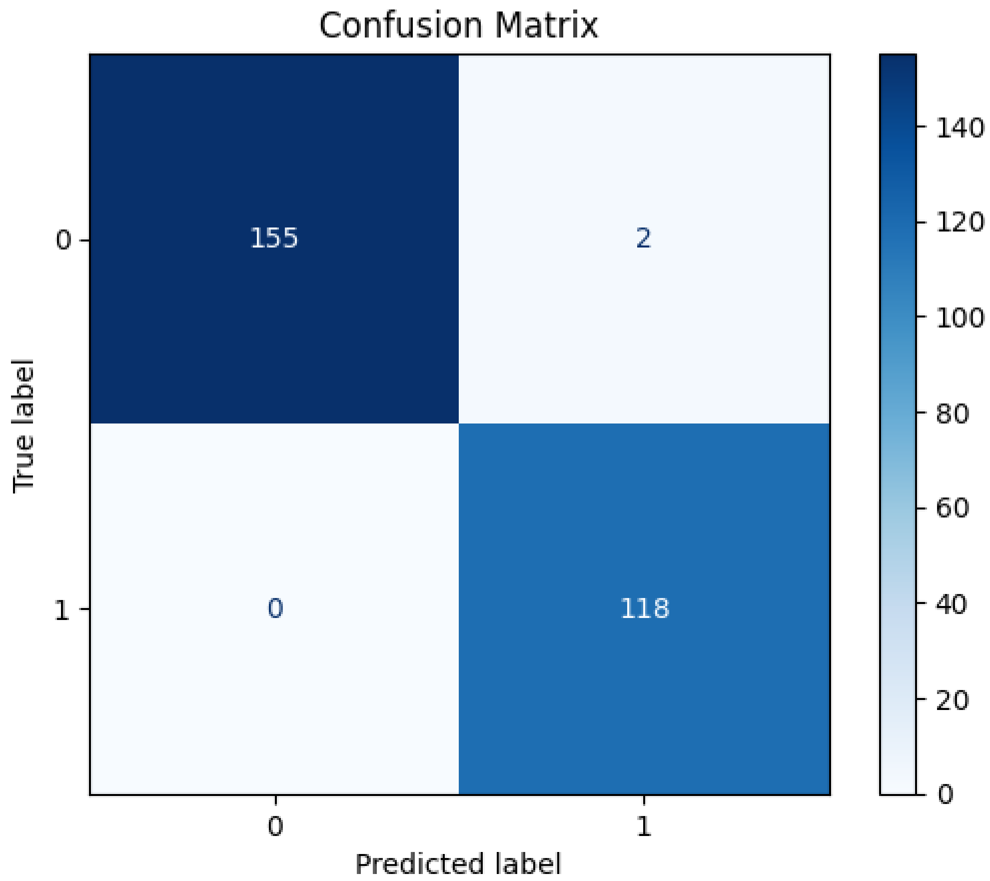


Figure-2

When I examined Figure-2, I observed that decision tree gave correct results for these values. Because F1, precision, accuracy and recall values are almost close to 1.

To make the model more complex, I set "max_depth" to 20 and "criterion" to 'entropy'. I left "min_samples_split" at the default value. Classification report, accuracy and confusion matrix are shown in Figure-3 for these values.

```
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.98      0.99       157
           1       0.97      0.99      0.98       118

    accuracy                           0.99       275
   macro avg       0.98      0.99      0.99       275
weighted avg       0.99      0.99      0.99       275

Accuracy: 0.9855
```

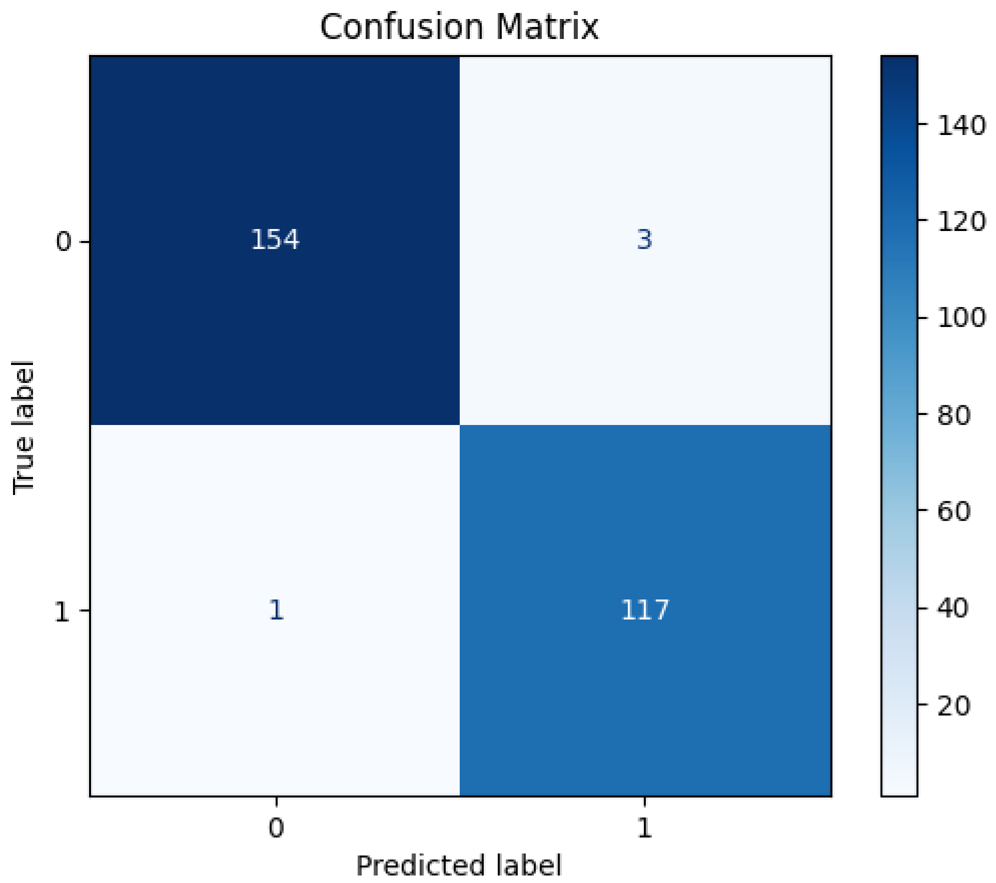


Figure-3

I observed that when I made the model more complex and deeper, accuracy, precision and recall values decreased.

As a next step, I tested the model in a way that significantly reduced its complexity and depth. I set "max_depth" to 3, "criterion" to 'gini' and "min_samples_split" to 10. Classification report, accuracy and confusion matrix are shown in Figure-4 for these values.

```
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.94      0.93       157
           1       0.92      0.90      0.91       118

    accuracy                           0.92       275
   macro avg       0.92      0.92      0.92       275
weighted avg       0.92      0.92      0.92       275

Accuracy: 0.9236
```

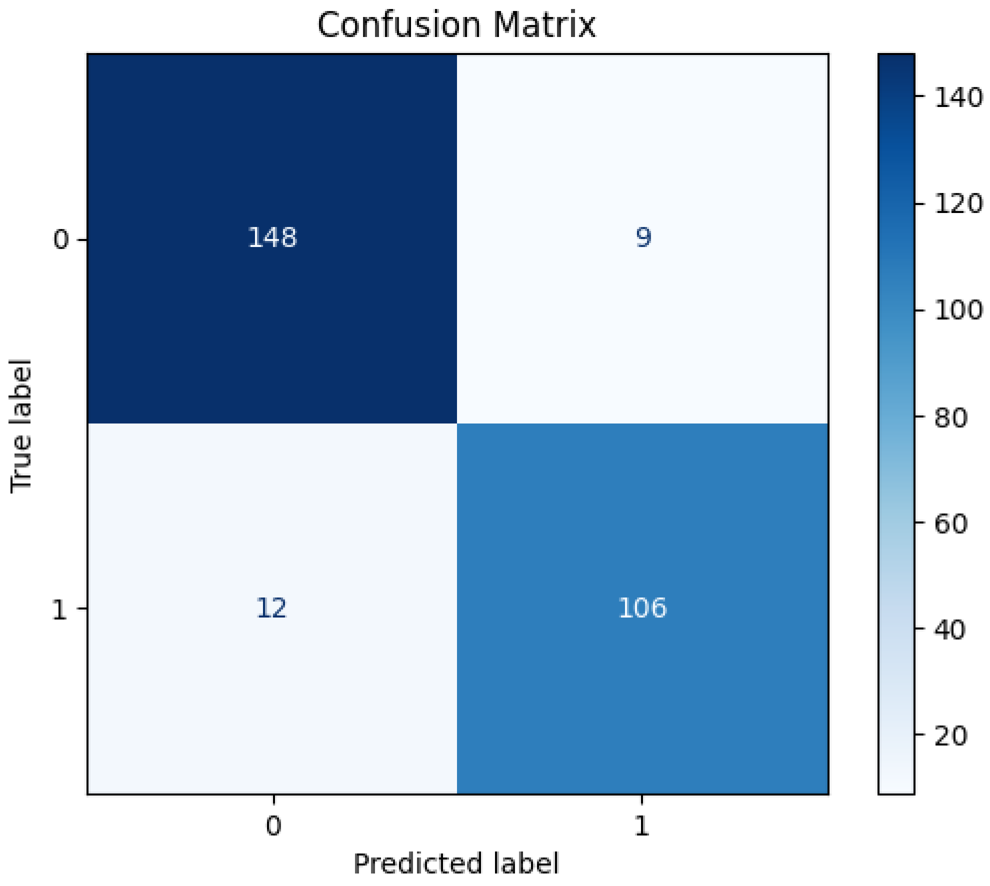


Figure-4

When I examined Figure-4, I saw that the accuracy of the model had decreased. For this reason, I repeated the test with new values to optimize the complexity and depth of the model and increase its accuracy.

Result for "max_depth" to 6, "criterion" to 'gini' and "min_samples_split" to 7 is shown in Figure-5. These values are optimum for tree.

```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.99      0.99       157
           1       0.98      1.00      0.99       118

    accuracy                           0.99       275
   macro avg       0.99      0.99      0.99       275
weighted avg       0.99      0.99      0.99       275

Accuracy: 0.9927
```

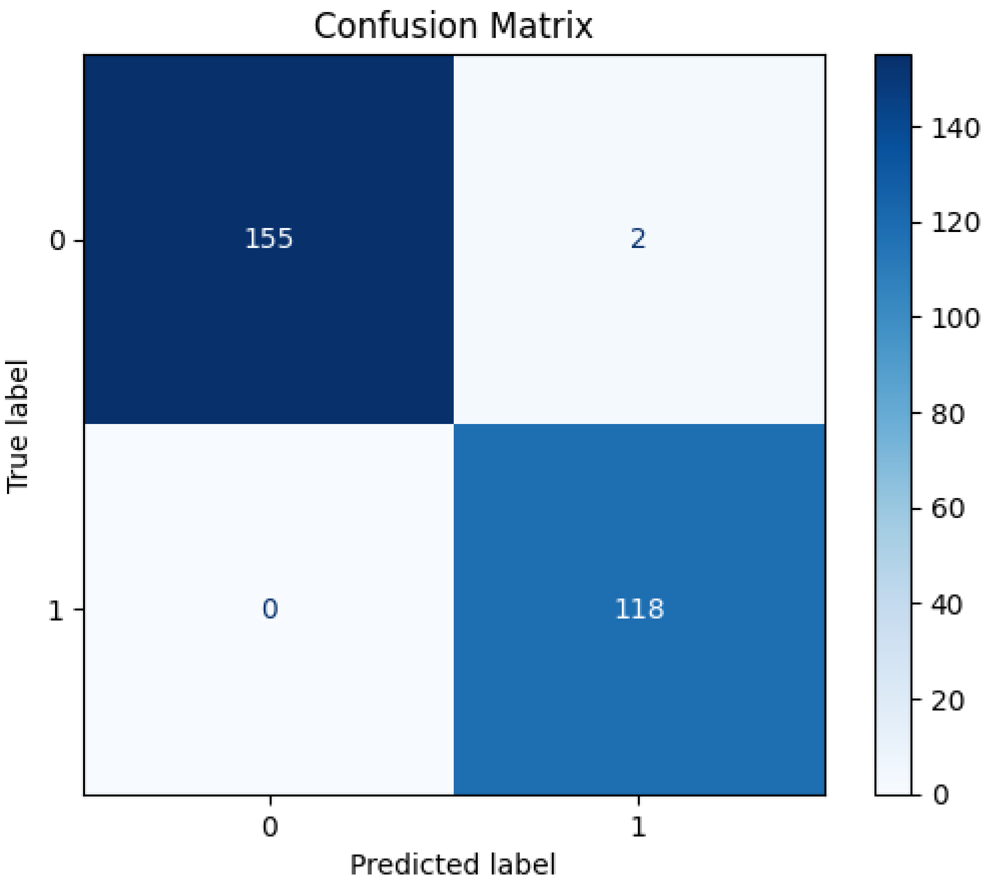


Figure-5

Result for "max_depth" to 5, "criterion" to 'gini' and "min_samples_split" to 15 is shown in Figure-6. Accuracy is decreased for these parameters.

```
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.99      0.97       157
           1       0.98      0.95      0.97       118

    accuracy                           0.97       275
   macro avg       0.97      0.97      0.97       275
weighted avg       0.97      0.97      0.97       275

Accuracy: 0.9709
```

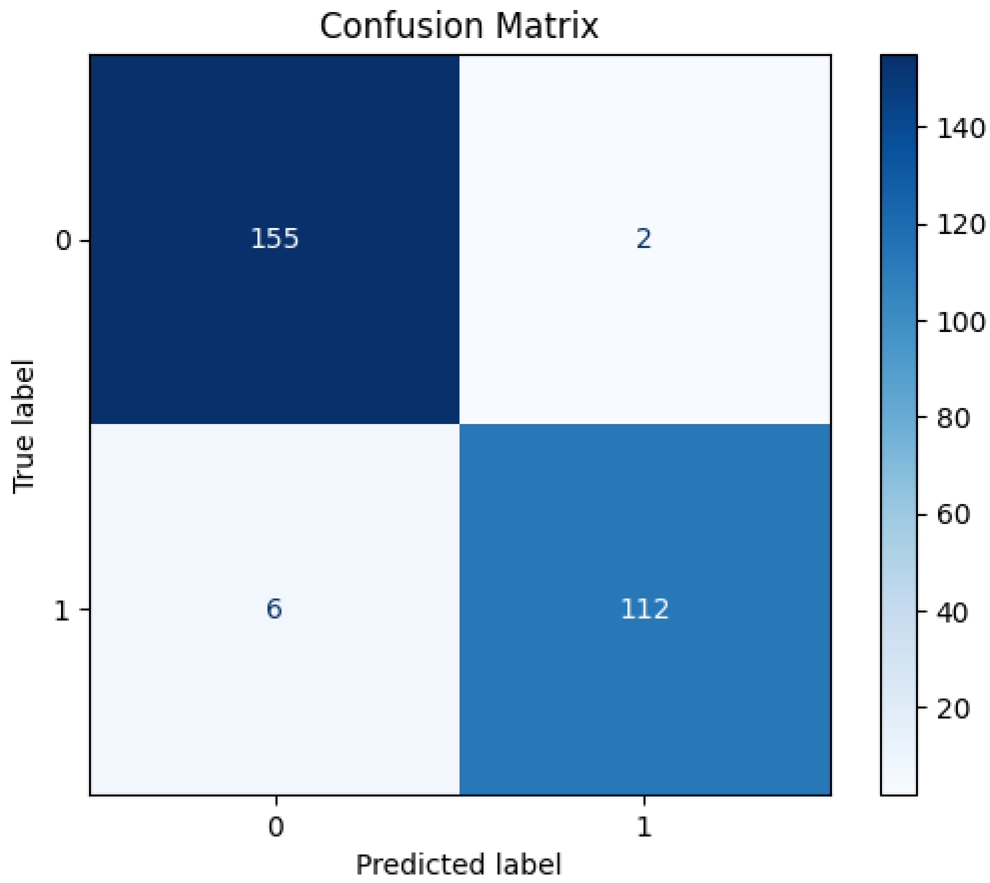


Figure-6

To observe the accuracy and complexity of the model, I created the model for different "max_depth" values and observed the results. While changing the "max_depth" values, I set "criterion" to 'gini' and "min_samples_split" to 7 and did not change it. The decision tree models created for different "max_depth" values are given below.

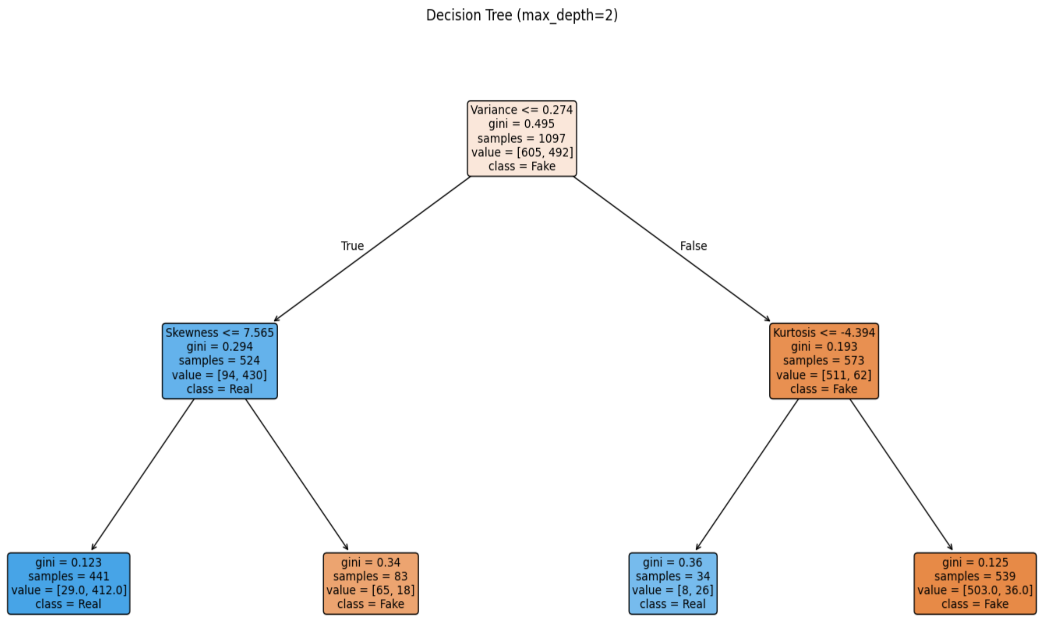


Figure-7

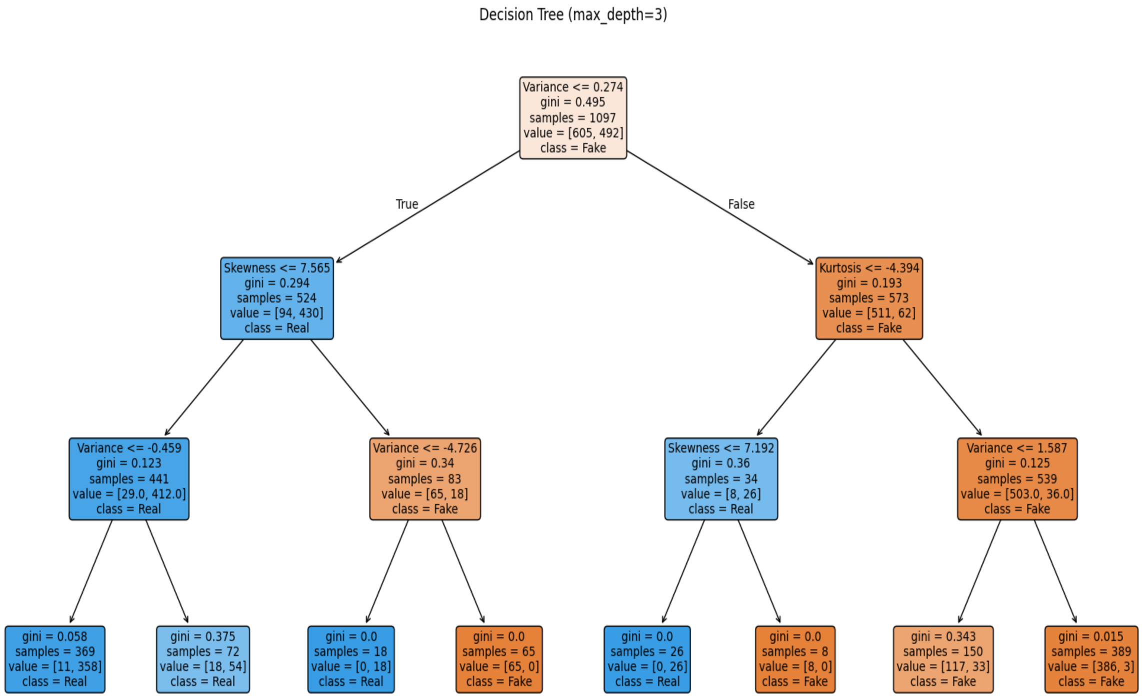


Figure-8

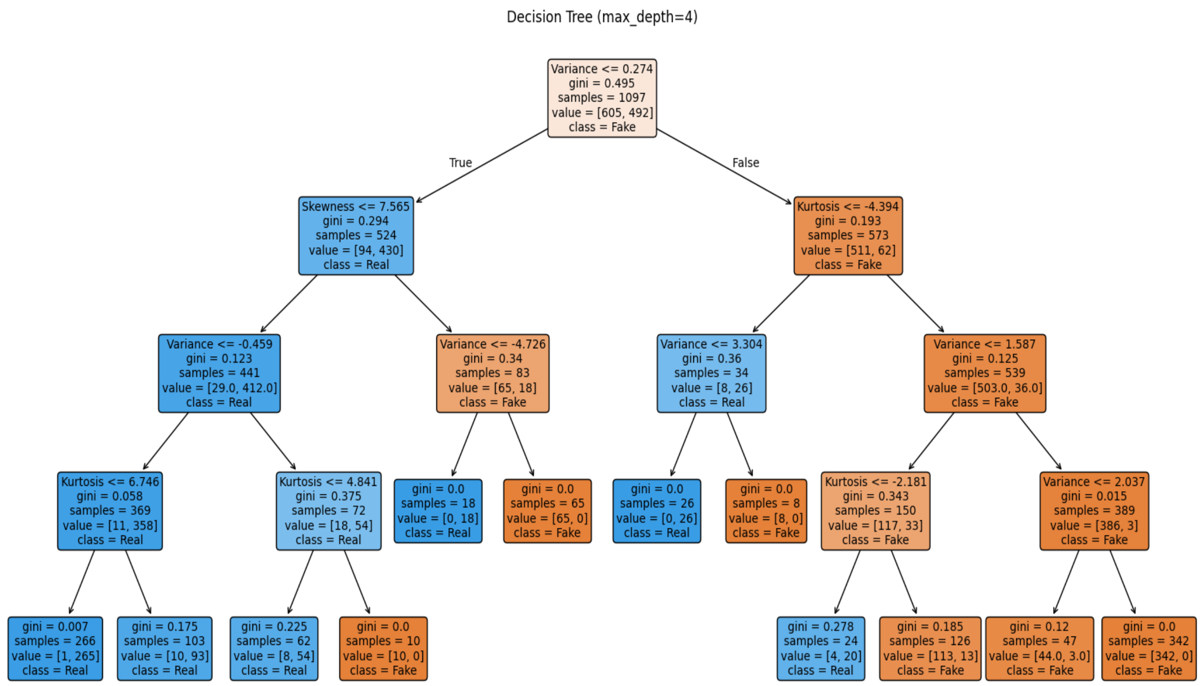


Figure-9

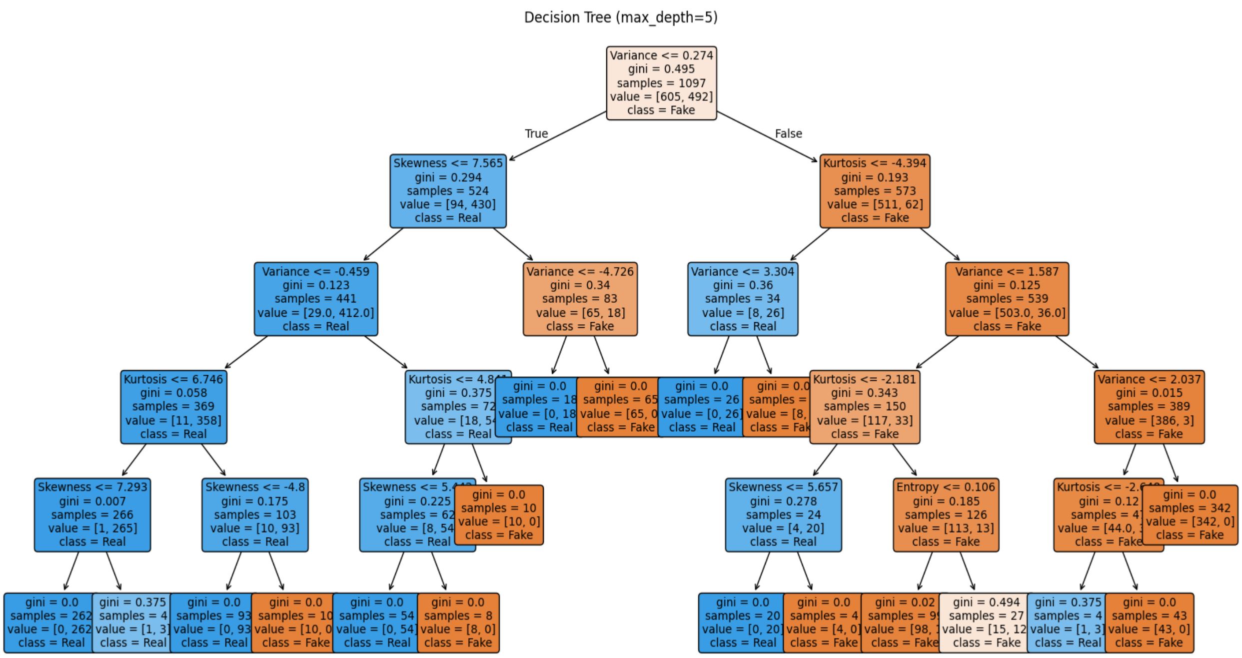


Figure-10

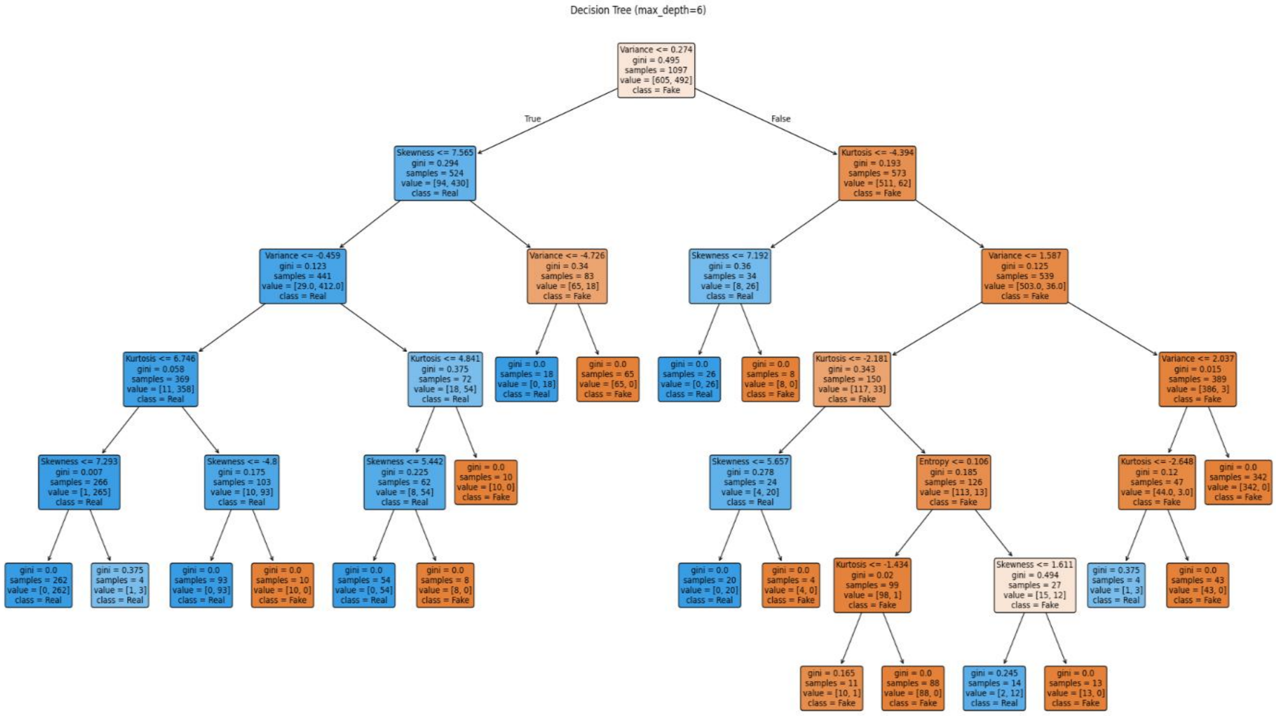


Figure-11

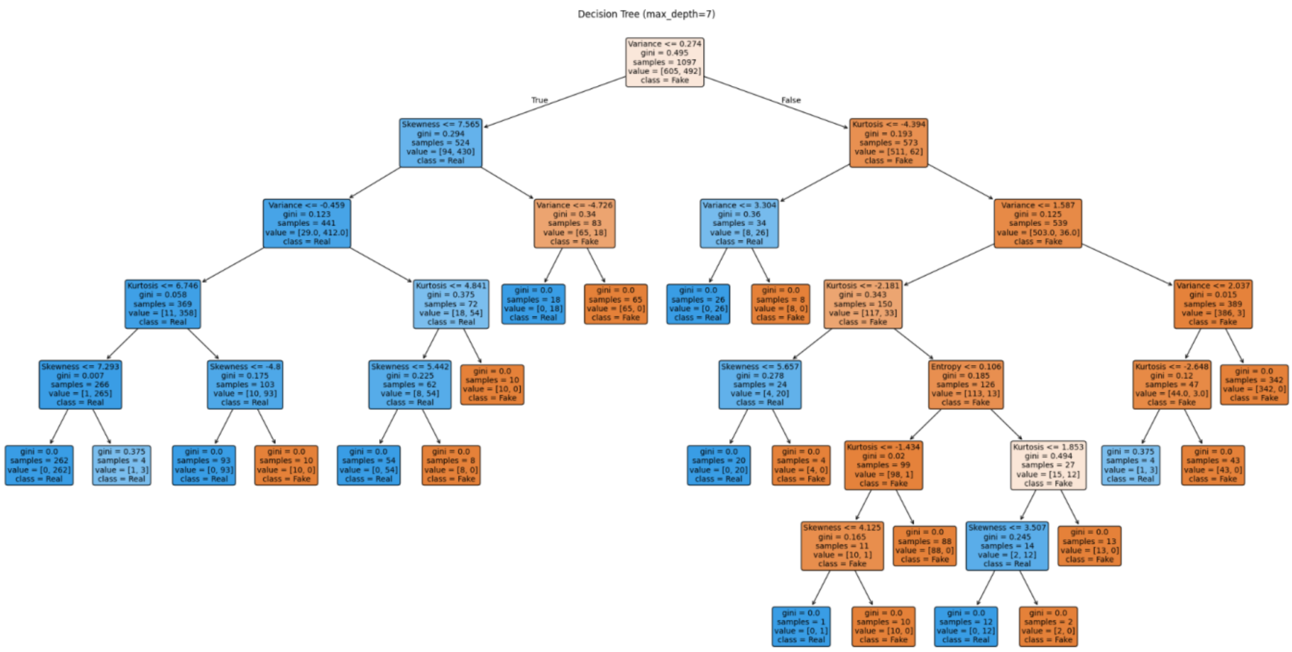


Figure-12

When the depth of the tree is increased, the complexity of the model increases, but the accuracy of the model also increases. In other words, there is a trade-off between the complexity and accuracy of the model. When the depth of the model is set to 2, 3 and 4, the accuracy of the model is low, but its complexity is low. When the depth of the model is set to

5, it starts to give accurate results, but when the depth of the model is set to 6 or 7, we build a much more accurate model. For this reason, choosing the depth of the model as 6 allows us to build the most accurate model.

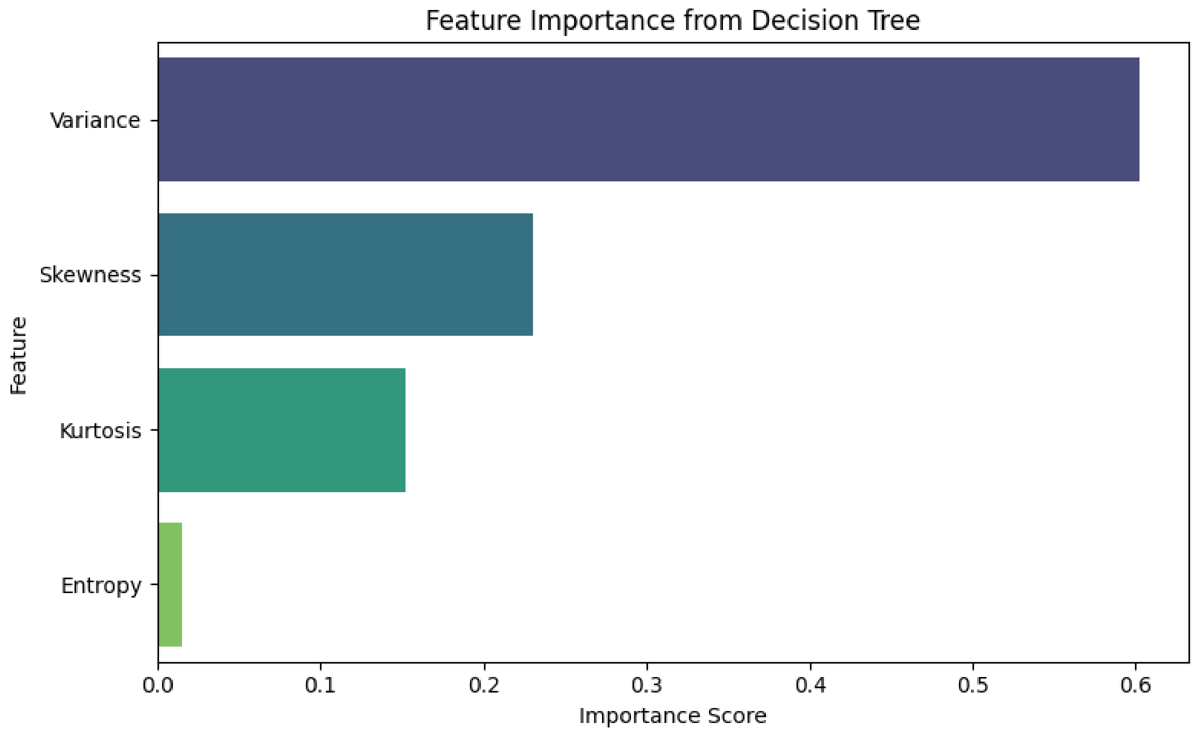


Figure-13

The most important feature used in creating the model is variance. For this reason, we see variance at the root of the decision tree. The model first separates the first samples according to the variance value. The other two important features are skewness and kurtosis. However, the discrimination of these two features is less than variance. The feature with the lowest importance score is entropy. Since the discrimination of this feature is quite low, it is not used in separating any nodes.

In this question, I learned that the size, complexity and accuracy of the model can be changed by changing parameters such as max_depth, min_samples_split, and criterion. I tried to find the optimum model by changing these values. I also compared the features with each other in pairs and observed which features were important in creating the model. I confirmed these observations in the "Feature Importance" graph.

I think that the decision tree model is a good model for this data set. Because by adjusting the mentioned parameters, the complexity and depth of the model can be adjusted and the model can be created to give accurate results. This method is also very good since we are working with a data set that is not very large and we want to observe how decisions are made graphically. However, if methods such as depth management or pruning are not used, it can overfit easily.

GitHub link: https://github.com/kivancates/decision-tree-ele489/tree/main