Developing a High-Precision Machine Learning Model for Mushroom Edibility Classification Based on Physical Features

Josh Austin Mikhail T. Natividad[1], Alexandra G. Padua[2], Lloyd Dominic G. Refuerzo[3], Joshua Emmanuel G. Tipon[4], Christian Dave P. Tordillo[5]

De La Salle University – Laguna Campus, 727V+352, LTI Spine Road, Laguna Blvd, Biñan, Laguna, Philippines

[1]austin_natividad@dlsu.edu.ph

[2]alexandra_padua@dlsu.edu.ph

[3]lloyd_refuerzo@dlsu.edu.ph

[4]joshua_emmanuel_g_tipon@dlsu.edu.ph

[5]christian_tordillo@dlsu.edu.ph

Abstract: While mushrooms are regularly used as an important ingredient in the culinary fields, some mushrooms can be poisonous. The paper aimed to develop a machine learning artificial intelligence that could classify the edibility of mushrooms based on their physical features. Two models were used, namely the Decision Tree Classifier and the Multilayer Perceptron neural network. Hyperparameter optimization was used to develop a model with the best accuracy and precision. Based on the results of each optimal model's performance metrics, it was concluded that the Decision Tree Classifier was best suited for this task.

Keywords: Mushroom Classification, Decision Tree Classification, Neural Network Classification, Multilayer Perceptron, Hyperparameter Optimization

1 Introduction

This paper aims to answer the question of what would be the best suitable machine learning algorithm to train an AI for classifying mushrooms based on physical characteristics to determine and classify combinations of these physical characteristics as poisonous or edible mushrooms.

Mushrooms are fungi, which belong in their own kingdom. They do not belong to the animal kingdom, as well as the plant kingdom. Their typical method of obtaining energy and nutrients differ from species from these other kingdoms as they use mycelium to grow into food sources. They use these mycelia to secrete the enzymes necessary for digestion, which then get extracted by the mycelium. The mushrooms are the reproductive organs of fungi, whereas compared to the fruits and seeds of a plant, these mushrooms release spores that are transported by factors such as air and insects. This method of germination allows the spores to form new networks of mycelia whenever these land on food sources [1].

Nowadays, indoor mushroom cultivation, also known as Fungiculture, is a very developed field of farming and research where instead of foraging for mushrooms in areas like forests, the edible mushrooms that are regularly found in supermarkets are now mass-produced to be used mostly in culinary [2]. There is even comprehensive information available today about starting mushroom cultivation, as some farming companies offer online courses to excel in this field.

However, while there are mushrooms that are regularly eaten as they are edible, some mushrooms, when ingested, could be poisonous and lead to food poisoning, or even death. There are some mushrooms such as the *Amina bisporigera,* commonly known as the “Destroying Angel”, that produce a poison wherein the ingester could develop symptoms of vomiting, diarrhea, and cramps after six to 24 hours before ingestion [3]. More lethal symptoms could be experienced later, such as kidney or liver dysfunction, which could result in death. This evolutionary trait of producing poison could be the reaction to insects and snails that often ingest these mushrooms as a source of nutrition [4].

The identification of edible mushrooms is of great importance as the avoidance of these poisonous mushrooms is much preferred in order to evade the consequences of ingesting poisons that could potentially lead to sickness or death. In Fungiculture, the cultivators must work in a controlled and sanitized environment to prevent the facilities from turning into breeding grounds for pests and mushrooms that were not meant to grow and eventually invade the environment [5]. Normally, these invasive fungi are not harmful and can be handled by the cultivator through pasteurization; however, precautions should be taken seriously by the cultivators.

By dealing with the production of crops that are regularly consumed, they must be aware of the mushroom pests that may be poisonous. This AI could be used as a tool for these cultivators to determine if a foreign mushroom species that grew into their facilities were to be poisonous or not, so they could proceed to perform the necessary actions in dealing with the mushroom pests. Aside from these, developing an AI that could determine the edibility of a mushroom could be useful for individuals who regularly forage these mushrooms. This paper aims to develop an AI that would possess the highest precision score as the model must not flag false positives at all, since they could lead to detrimental effects on a person's health.

2 Mushroom Classification Dataset

Created by UCI Machine Learning, the Mushroom Classification Dataset is a collection of information regarding the different types of mushrooms, containing over 8124 samples and 22 attributes. The attributes that are contained as the information present in the data set are about the shape, color, odor, the mushroom’s cap, gills, stalk, and veil, as well as if the mushroom possesses bruises, and the type of spore print it releases. Each of these attributes has a

limited number of possible values. Its features provide useful information for predicting the class label, which makes it an excellent tool.

Listing 1. Features and attributes found in the Mushroom Classification Dataset.

- Cap-Shape:
  - Describes the shape of the mushroom's cap.
    - 'b' - bell
    - 'c' - conical
    - 'x' - convex
    - 'f' - flat
    - 'k' - knobbed
    - 's' - sunken
- Cap-Surface:
  - Describes the surface of the mushroom's cap.
    - 'f' - fibrous
    - 'g' - grooves
    - 'y' - scaly
    - 's' - smooth
- Cap-Color:
  - Describes the color of the mushroom's cap.
    - 'n' - brown
    - 'b' - buff
    - 'c' - cinnamon
    - 'g' - gray
    - 'r' - green
    - 'p' - pink
    - 'u' - purple
    - 'e' - red
    - 'w' - white
    - 'y' - yellow
- Bruises:
  - Describes whether the mushroom has bruises or not.
    - 't' - bruises
    - 'f' - no bruises
- Odor:
  - Describes the odor of the mushroom.
    - 'a' - almond
    - 'i' - anise
    - 'c' - creosote
    - 'y' - fishy
    - 'f' - foul
    - 'm' - musty
    - 'n' - none
    - 'p' - pungent
    - 's' - spicy
- Gill-Attachment:
  - Describes how the mushroom's gills are attached to the stem.
    - 'a' - attached

        - ‘f’ - free
- Gill-Spacing:
    - Describes the spacing between the mushroom’s gills.
        - ‘c’ - close
        - ‘w’ - crowded
- Gill-Size:
    - Describes the size of the mushroom’s gills.
        - ‘n’ - narrow
        - ‘b’ - broad
- Gill-Color:
    - Describes the color of the mushroom’s gills.
        - ‘k’ - black
        - ‘n’ - brown
        - ‘b’ - buff
        - ‘h’ - chocolate
        - ‘g’ - gray
        - ‘r’ - green
        - ‘o’ - orange
        - ‘p’ - pink
        - ‘u’ - purple
        - ‘e’ - red
        - ‘w’ - white
        - ‘y’ - yellow
- Stalk-Shape:
    - Describes the shape of the mushroom’s stalk
        - ‘e’ - enlarging
        - ‘t’ - tampering
- Stalk-Root:
    - Describes the type of root that the mushroom’s stalk has.
        - ‘b’ - bulbous
        - ‘c’ - club
        - ‘u’ - cup
        - ‘e’ - equal
- Stalk-Surface-Above-Ring:
    - Describes the surface of the mushroom’s stalk above the ring
        - ‘f’ - fibrous
        - ‘g’ - grooves
        - ‘y’ - scaly
        - ‘s’ - smooth
- Stalk-Surface-Below-Ring:
    - Describes the surface of the mushroom’s stalk below the ring
        - ‘f’ - fibrous
        - ‘g’ - grooves
        - ‘y’ - scaly
        - ‘s’ - smooth
- Stalk-Color-Above-Ring:
    - Describes the color of the mushroom’s stalk above the ring
        - ‘n’ - brown

        - ‘b’ - buff
        - ‘c’ - cinnamon
        - ‘g’ - gray
        - ‘o’ - orange
        - ‘p’ - pink
        - ‘e’ - red
        - ‘w’ - white
        - ‘y’ - yellow
- Stalk-Color-Below-Ring:
    - Describes the color of the mushroom’s stalk below the ring
        - ‘n’ - brown
        - ‘b’ - buff
        - ‘c’ - cinnamon
        - ‘g’ - gray
        - ‘o’ - orange
        - ‘p’ - pink
        - ‘e’ - red
        - ‘w’ - white
        - ‘y’ - yellow
- Veil-Type:
    - Describes the type of veil that covers the mushroom’s gills
        - ‘p’ - partial
- Veil-Color:
    - Describes the color of the mushroom’s veil
        - ‘n’ - brown
        - ‘o’ - orange
        - ‘w’ - white
        - ‘y’ - yellow
- Ring-Number:
    - Describes the number of rungs that the mushroom has.
        - ‘n’ - none
        - ‘o’ - one
        - ‘t’ - two
- Ring-Type:
    - Describes the type of ring that the mushroom has.
        - ‘c’ - cobwebby
        - ‘e’ - evanescent
        - ‘f’ - flaring
        - ‘l’ - large
        - ‘p’ - pendant
- Spore-Print-Color:
    - Describes the color of the mushroom’s spore print
        - ‘b’ - black
        - ‘n’ - brown
        - ‘h’ - chocolate
        - ‘r’ - green
        - ‘o’ - orange
        - ‘u’ - purple

        - ‘w’ - white
        - ‘y’ - yellow
- ➢ Population:
    - Describes the abundance of the mushroom
        - ‘a’ - abundant
        - ‘c’ - clustered
        - ‘n’ - numerous
        - ‘s’ - scattered
        - ‘v’ - several
        - ‘y’ - solitary
- ➢ Habitat
    - Describes where the mushroom is found
        - ‘g’ - grasses
        - ‘l’ - leaves
        - ‘m’ - meadows
        - ‘p’ - paths
        - ‘u’ - urban
        - ‘w’ - waste
        - ‘d’ - woods

Listing 2. Class Labels found in the Mushroom Classification Dataset

- ➢ Classes: Indicates whether a mushroom is edible or poisonous
    - ‘e’ - edible
    - ‘p’ - poisonous

2.1 Class Distribution

The distribution of the classes is relatively balanced, with only a somewhat higher proportion of edible mushrooms being roughly 4.4% higher than poisonous mushrooms.

- ➢ Class ‘e’ - edible mushrooms
    - 52.2%
- ➢ Class ‘p’ - poisonous mushrooms
    - 47.8%

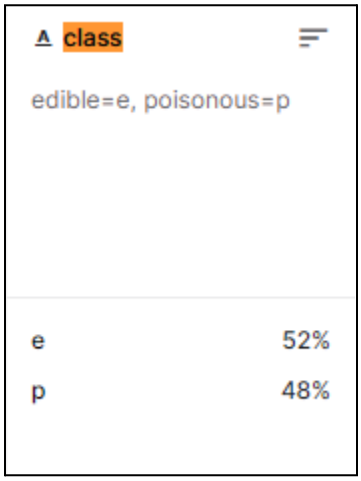

Fig. 1. Class Distribution taken from Kaggle.com

3 Methodology

3.1 Data Preprocessing

To create the models for both the Decision Tree Classification model and the Neural Network model, a predictive machine learning pipeline was followed which involved the steps of preprocessing, learning, evaluation, and prediction. The Mushroom Classification model was retrieved from kaggle.com [6].

The categorical variables were converted into numerical data using the Pandas Library get_dummies function. This creates new binary columns for each unique label and uses a 0 or 1 to indicate which category is present in raw data. For example, in the original dataset, to describe the cap shape of a mushroom, the following character values are seen in the table below.

Table 1. List of characteristics of a mushroom's cap shape in character symbol format.

Cap-shape
b
c
x
f
k
s

This type of data may require more steps for the machine learning algorithms to use; therefore, dummy encoding was used to convert it into numerical data, as seen in an example table below. For each feature, a row would only have a binary column that is set to 1, while the rest is 0 to indicate the presence of the feature in the sample. This type of encoding was applied to every attribute.

Table 2. Dummy encoded characteristics of a mushroom's cap shape.

Cap-shape_b	Cap-shape_c	Cap-shape_x	Cap-shape_f	Cap-shape_k	Cap-shape_s
1	0	0	0	0	0
0	1	0	0	0	0
0	0	1	0	0	0
0	0	0	1	0	0
0	0	0	0	1	0
0	0	0	0	0	1

Once the mushroom classification dataset was dummy encoded, the dataset was split into its training and test datasets, using the train_test_split function provided by sci-kit learn. The group decided to use 70% of the dataset as its training set, while the rest would be for testing. By using the random_state parameter, the dataset was randomized on whether it would be split into the training or test dataset.

3.2 Decision Tree Classifier Model Learning, Evaluation, and Prediction

After preparing the training and test dataset, the learning part of the pipeline was started as the data that would be used to train the models are now prepared. For the first model, the group used Scikitlearn's Decision Tree Classifier model which would build a tree of decisions that could classify based on the training data. For this model, the hyperparameter tuning would revolve around two hyperparameters, the decision tree classifier's max depth, and its criterion. The max depth parameter of the model controls the size of the decision tree as it would limit the creation of the tree to a specific size. This hyperparameter has a significant effect on how the model would perform, as a higher maximum depth would result in a more complex tree with more decisions that are considered; however, overfitting could be a problem if the decision tree were given a very high value for the maximum depth. Having a smaller tree due to a limited maximum depth would increase the model's explainability, as it would be easier to trace a model that only has a few amounts of decisions that are made. These effects were noted by the group to choose the best decision tree model that was created.

Another hyperparameter would be the measurement of the randomness of a set, where two metrics were compared, namely the Gini index and Entropy. The Gini index is obtained by subtracting the sum of the squared probabilities of each class from the value 1. This metric is more focused on the probability of a certain randomly [8]selected feature that was incorrectly classified [7]. In machine learning, entropy refers to the randomness of measuring the impurity of information that is processed by a decision tree. This measures the unpredictability of the data given that would be processed by the machine learning algorithm [8]. These metrics are important as the two could provide different performance metric scores since the functionality and equations of the two metrics differ greatly, as the Gini index only uses arithmetic, while entropy deals with logarithms of base 2.

The hyperparameters would be changed according to the performance metrics of the model at that tuning. The group tested both the Gini index and the entropy metric with varying maximum tree depths, starting from 2 until a good performance was achieved on the evaluation using the test set. The group aimed for a decision tree with a high-accuracy model, as well as a high precision score, as it would be important for the model to avoid predicting false positives. Once the best model was determined, it was tested with new data to observe if the model was able to correctly classify the data.

3.3 Neural Network Model Learning, Evaluation, and Prediction

The group employed the Multilayer Perceptron Model as our neural network model for the project. It is a powerful neural network model in the field of machine learning. It is well-suited for tasks such as picture and audio recognition, natural language processing, and many other applications because of its ability to handle complex nonlinear interactions between inputs and outputs. The success of the MLP model can be attributed to its architecture, which consists of multiple layers of connected nodes or neurons. Each layer of the MLP transforms the input data differently, allowing the model to learn from the data and extract higher-level features and patterns. This layered approach enables the MLP to model complex functions with greater accuracy and generalization ability while avoiding overfitting, a common problem in machine learning. Overall, the Multilayer Perceptron Model is an essential tool for building intelligent systems that can perform a variety of complex tasks with high accuracy and efficiency.

The hyperparameters used were the default with the max number of iterations at 400 and the random state at 42. The hyperparameter that was changed to use for the comparison of the models was the activation functions. The group used 4 activation functions for comparison namely: ReLU, Identity, Logistic, and tanh activation functions.

The ReLU activation function is computationally efficient and helps prevent the vanishing gradient problem in deep neural networks, making it a popular choice. However, ReLU may cause dead neurons or result in a

gradient that is too large, leading to unstable training. The Identity activation function is simple, and its output equals its input, making it useful for tasks like regression. However, it lacks nonlinearity, which can limit the network's ability to learn complex patterns. The Logistic activation function is commonly used in binary classification problems and provides an output that is bounded between 0 and 1, which can be interpreted as a probability. However, it has a small gradient for extreme inputs, which can cause slow learning and saturation. The tanh activation function is similar to the logistic function but outputs values between -1 and 1. It is often used in multi-class classification problems and can provide stronger nonlinearity than the logistic function. However, it is also prone to saturation for extreme inputs, which can cause slow learning and vanishing gradients.

In summary, choosing the appropriate activation function depends on the task at hand and the network architecture. The ReLU function is efficient but can lead to dead neurons, while the Identity function lacks nonlinearity. The Logistic function is useful for binary classification but can suffer from slow learning, while the tanh function can provide strong nonlinearity but is prone to saturation.

4 Results and Analysis

4.1 Decision Tree Classification Technique

The model was tested in different hyperparameters, and varying results were observed from each test. While changing the maximum depth of the decision tree model under the entropy metric, it was observed that the model could already classify the dataset with a maximum depth of 5, as seen in the figure below.

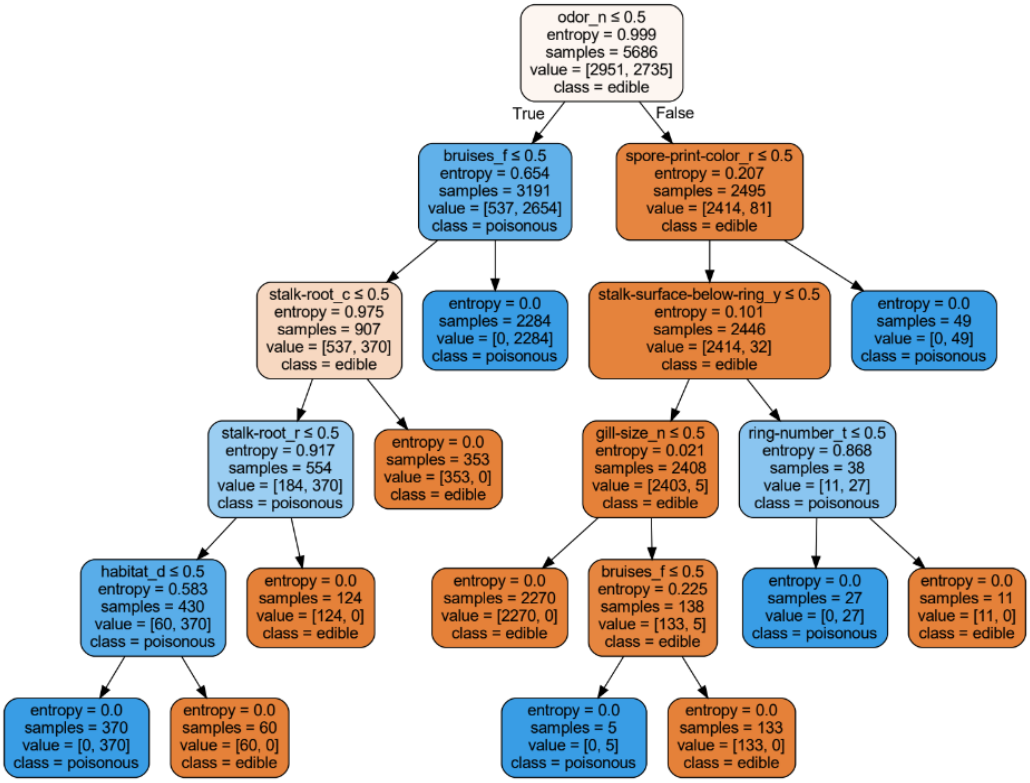

Fig. 2. Visualization of the decision tree classification model with a maximum depth of 5 and the entropy metric.

The decision tree model can already determine the edibility of a mushroom as the tree has already classified the entire dataset into the two labels using decisions on whether a mushroom would exhibit this characteristic or not. As

seen in the table below, the performance metric scores of the training set were already high, especially the precision of the model scored 99.97%. The group decided to evaluate this model as the best model, where it was able to correctly classify all of the data added to the test set. The accuracy of the decision tree model seems to increase as the maximum depth of the model is increased, mainly due to the availability of more features the decision tree model is able to use to further split the dataset.

Table 3. Performance metric scores of training and test set of a decision tree classifier with an entropy metric for different maximum depths.

HYPERPARAMETERS		TRAINING SET				TEST SET			
MAX_DEPTH	CRITERION	ACCURACY	PRECISION	RECALL	F1	ACCURACY	PRECISION	RECALL	F1
2	Entropy	0.9293000352	0.9400536833	1	0.9362309645	0.9220672683	0.9343469247	1	0.9297337278
3	Entropy	0.9648258881	0.9657611976	0.9339207048	0.9649859944	0.9540607055	0.9563502431	0.9132856006	0.9534883721
4	Entropy	0.9887442842	0.9885603182	0.9796679092	0.9890523435	0.9840032814	0.9839474917	0.9713603819	0.984280532
5	Entropy	0.9996482589	0.9996613613	1	0.9996612466	1	1	1	1

As seen in the table, the performance metrics, specifically the accuracy, precision, and f1 scores were low when the maximum depth is 2 since the decision tree was only able to classify with at most 7 features, which gave inaccurate results, as it can be seen that the accuracy is only equal to approximately 92% on both the training set and test set. However, once the maximum depth hyperparameter was increased, the performance metric scores showed improvement, visually seen in the graphs below.

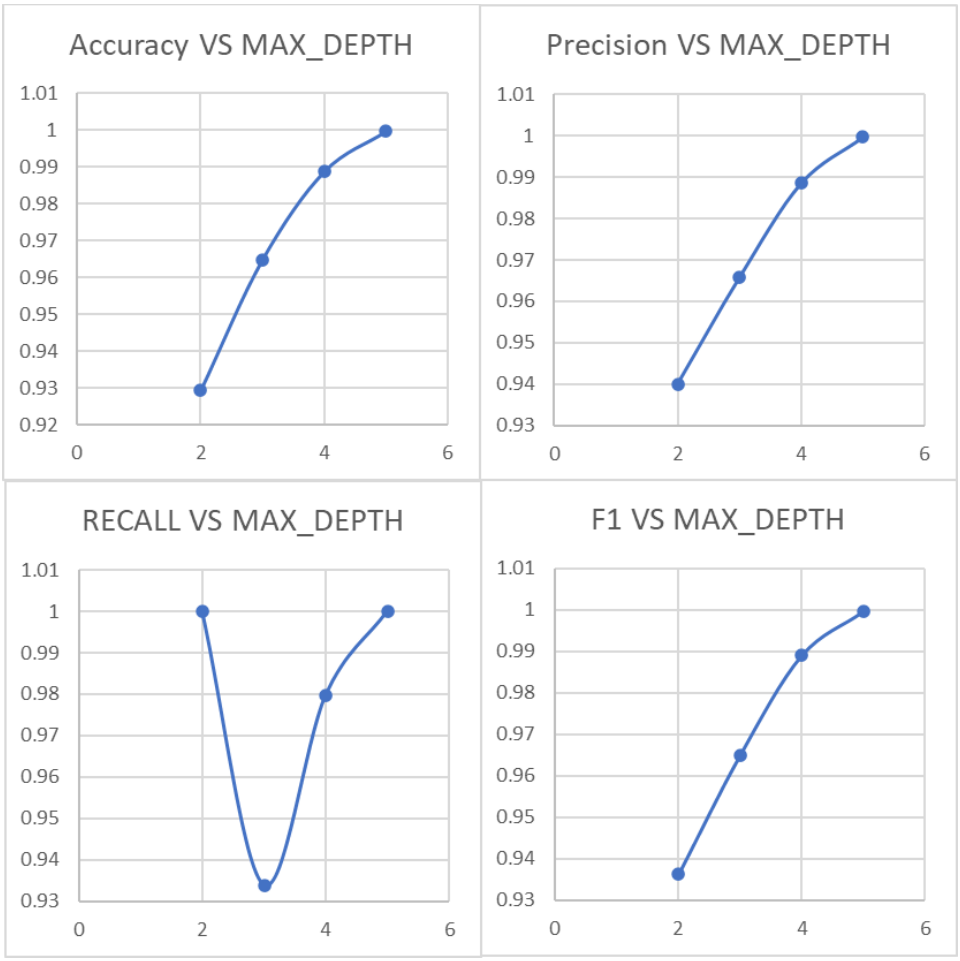

Fig. 3. Graph visualizations of the different performance metrics for the decision tree classifier model with an entropy metric.

When the model was tested using the Gini index metric, different results were observed. Compared to the model using an entropy metric, the decision tree was not able to classify the entire training data set at a maximum depth of 5. It can be observed in the table below that the performance of the model that uses the Gini index is slightly lower compared to the entropy index at a maximum depth of 5. The model was also subpar compared to the model in the testing set, as the model does not have 100% accuracy unlike the model using the entropy metric.

Table 4. Performance metric scores of training and test set of a decision tree classifier with a Gini metric for different maximum depths.

HYPERPARAMETERS		TRAINING SET				TEST SET			
MAX_DEPTH	CRITERION	ACCURACY	PRECISION	RECALL	F1	ACCURACY	PRECISION	RECALL	F1
2	Gini	0.956911713	0.95702465	0.9376482548	0.957605122	0.9479081214	0.949016651	0.9172633254	0.9478010686
3	Gini	0.9859303553	0.9857481333	0.9762792274	0.9863060596	0.98195242	0.98195242	0.9673826571	0.9822294023
4	Gini	0.9927893071	0.9926546851	0.9888173501	0.9930236515	0.9917965546	0.9916731794	0.984884646	0.9919871795
5	Gini	0.998417165	0.9984215876	0.9986445273	0.9984753515	0.9995898277	0.9996025437	1	0.9996023857

With all these in mind, the group decided to select the model with the hyperparameters of maximum depth 5 using the entropy metric. This is due to the fact that it was able to perform better than the model using a Gini index at the same maximum depth, as well as exhibiting the best performance metric scores in both the training set and data set while having the highest precision score as well. The comparison of the performance metric scores can be seen below and it shows that the model using the entropy metric produced the highest scores.

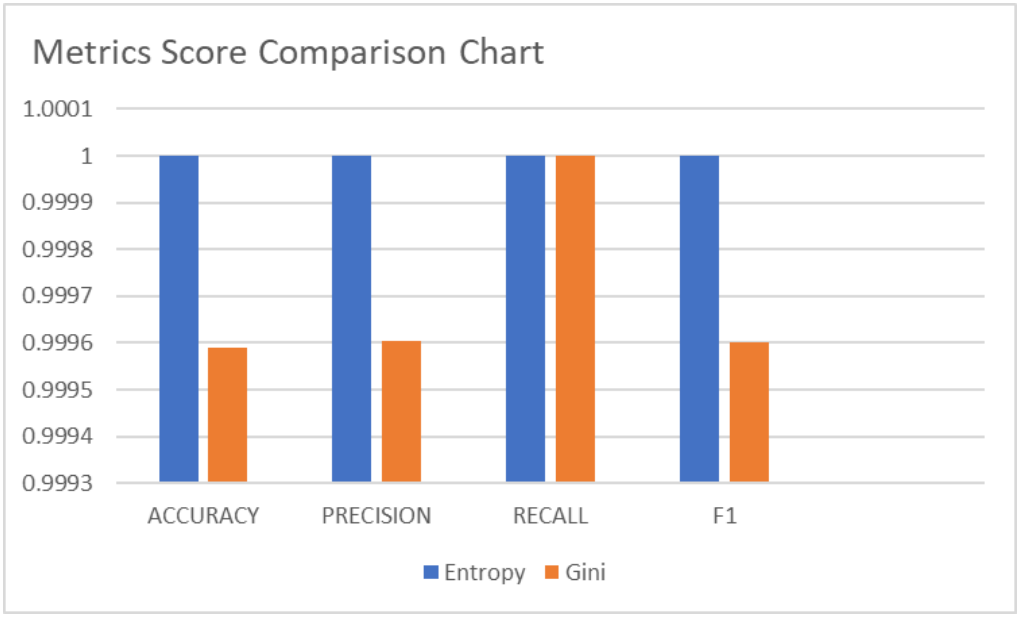

Fig. 4. Metric score comparison chart for the performance metric scores of the decision tree classifier model with an entropy metric and a Gini metric.

To further compare the results between the best model using the Gini index metric and the best model using the entropy metric, its confusion matrices were compared, as well as its cross-validation scores. Seen in the figures below are the results of the best models for each metric.

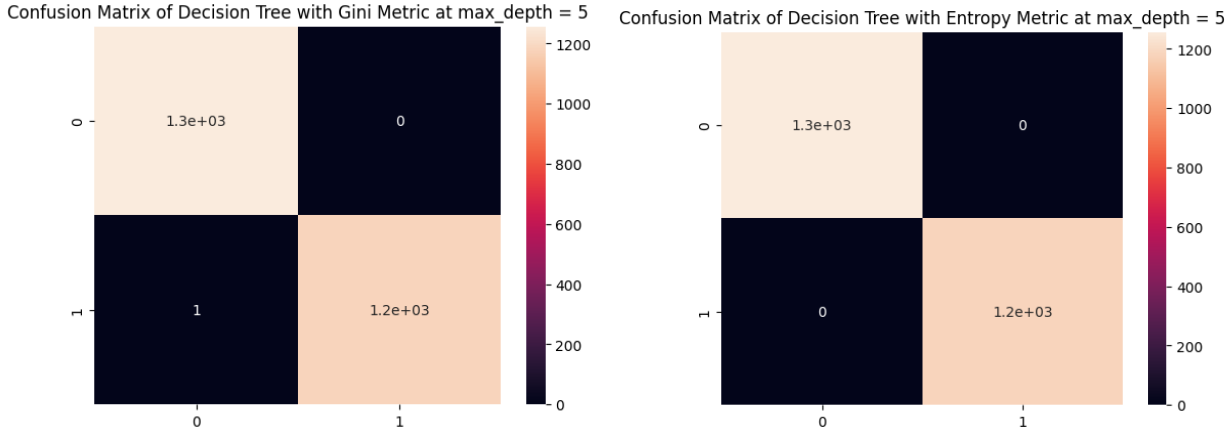

Fig. 5. Confusion matrices of the decision tree classifier model with an entropy metric and a Gini metric.

These confusion matrices are obtained from the test data set after the training of the decision tree classifier model. It can be observed that the decision tree model with the entropy metric performed better than the decision tree with the

Gini metric. The difference in performance between the two models was also observed when comparing the cross-validation accuracy scores. At the second k-fold of the model using the entropy metric, it was observed that the accuracy was already at 100% from this k-fold as well as the succeeding ones. However, for the Gini index, while it ended at a 100% accuracy score as well in the 5th fold, it was observed that it took a long time as the accuracy was around 99% in the first 4 folds of the cross-validation.

4.2 Neural Network Classification Technique

The model that was used is the Multilayer Perceptron for the neural network classification technique. To test which hyperparameters were optimal to use, the group evaluated the model using the four performance metrics. The results of the four different activation functions resulted in different metric scores. Provided below are the table and the chart comparing the metric scores of the MLP model under the four different hyperparameter activation functions.

Table 5. Metrics score table using ReLU, identity, logistic, and tanh activation functions

	ReLU	Identity	Logistic	tanh
Accuracy Score	1.00	0.9998241294407310	0.9996482588814630	0.9998241294407310
Precision Score	1.00	0.9998306233062330	0.9996613613274630	0.9998306233062330
Recall Score	1.00	1.00	1.00	1.00
F1 Score	1.00	0.9998305946129080	0.9996612466124660	0.9998305946129080

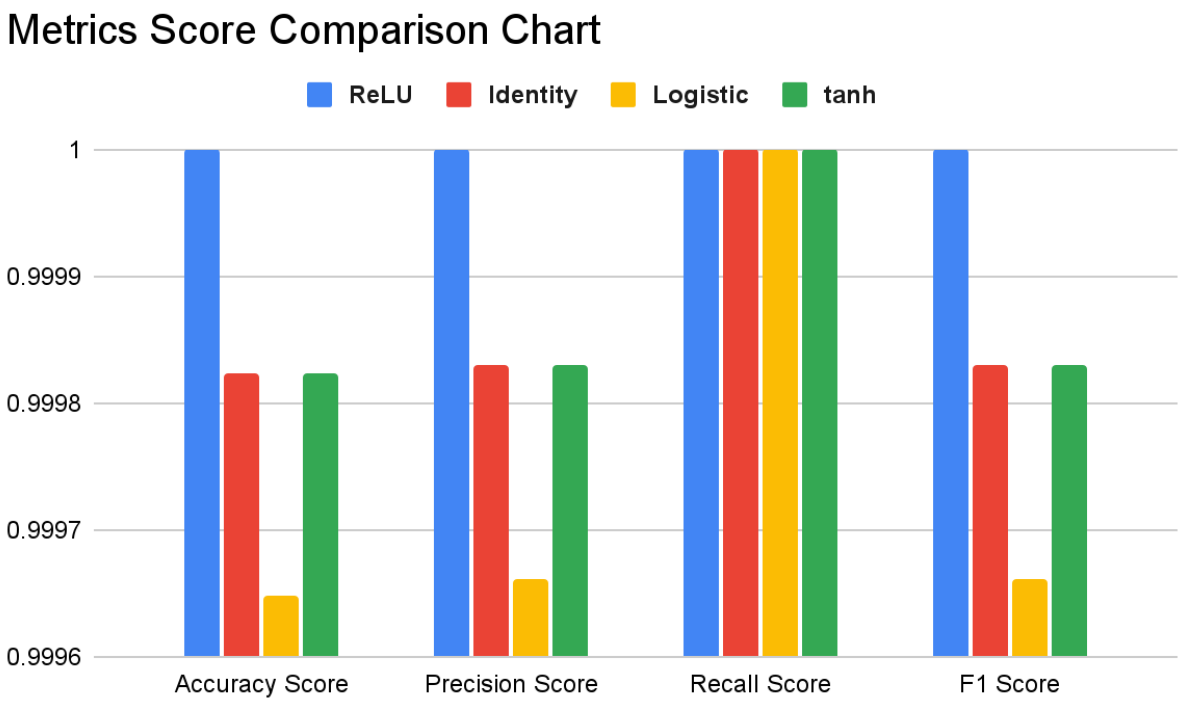

Fig. 6. Metrics score comparison chart for the different hyperparameters tested.

The results show that using the ReLu activation function gives the best scores among the four activation functions used meaning that it has achieved the perfect performance for the dataset used. This means that the classifier correctly predicted all instances of the positive class and did not give any false positives or false negatives. However, that may also mean that the model may have been overfitted due to a non-optimized number of iterations. The results also show that the metric scores for using the Identity and tanh activation functions are equal which might be attributed to their comparable characteristics. Despite having diverse results, each activation function has a complete recall, indicating that all of the acquired data is relevant.

It was therefore concluded that the ReLU activation function performed the best with perfect scores of 1.0 in all metrics hence, the optimal hyperparameters for this particular task would be to use the ReLU activation function. The Identity and tanh functions had very similar and high scores in all metrics, while the Logistic function had slightly lower scores.

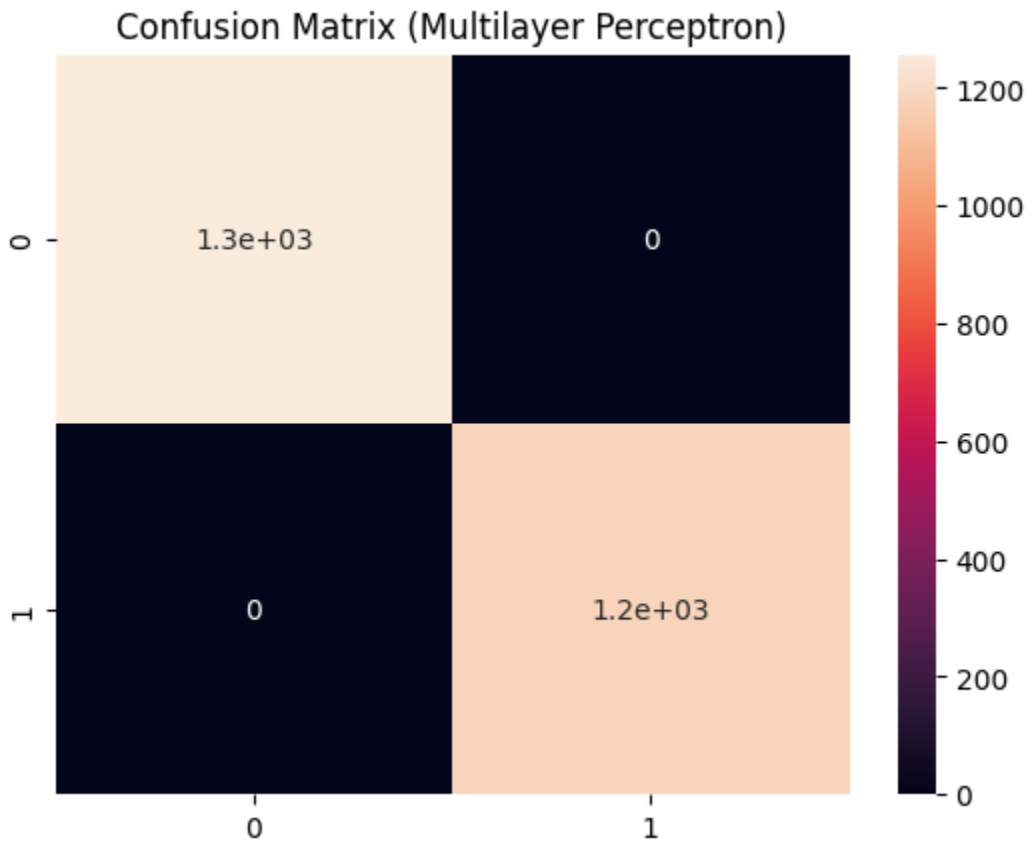

Fig. 7. Multilayer perceptron confusion matrix of the test set.

Based on these results, it was hypothesized that the ReLU function's ability to handle and model nonlinear interactions between the inputs and outputs effectively may have contributed to its superior performance. In contrast, the Identity and tanh functions may have performed well because they are good at modeling complex interactions between the inputs and outputs as well, but their nonlinear transformations are different from those of the ReLU function. As for the Logistic function, its S-shaped curve may have led to less effective modeling of the nonlinear interactions between the inputs and outputs, which may have resulted in slightly lower scores.

4.3 Comparison of Decision Tree Classification Technique and Neural Network Classification Technique

There are certain differences between the decision tree classification and the neural network classification. According to Andre Y., decision trees are more structured than a neural network's probability [9]. The decision tree approach reduces entropy, is considered deterministic, and explicitly fits the parameters of the conditions given. The neural network approaches the problem with the manipulation of shapes of activation functions, focuses on probability, and fits parameters to transform input and indirectly leads the activations of other neurons. As observed from the bar graphs, the entropy and ReLu functions have the same value of 1.0, upon observing the other four functions the neural network is more likely to be overfitted than the decision tree. The performance metrics of the neural network of four activation functions are higher than the performance metrics of the decision tree by 2% because the decision tree can only handle 7 features at max. Despite the outrageous performance metrics that the neural network provided, the group decided to lean towards the decision tree since it is much more practical since it is easier to interpret and understand due to the data being illustrated in a tree and using entropy as the basis for determining the order of the information given.

5 Conclusion and Recommendations

In summary, the objective of this paper is to determine the best suitable machine learning algorithm for training artificial intelligence, which will then be able to classify mushrooms based on a given set of different characteristics and other criteria, as well as be able to identify if the specified mushroom poses a risk to the health and safety of people or it is safe to eat.

It has been determined that the best model for the task would be the Decision Tree Classifier, as it was observed to be able to classify the test set with 100% accuracy, while being explainable as the decision tree can be visually rendered and explained how the logic behind the trained model works. While the neural network machine learning algorithm was able to provide similar results, the explainability behind the model is quite low. There is also an observation that the neural network is more likely to be overfitted.

The performance of the model is optimal and suitable for classifying the edibility of a mushroom given its physical features, as it was able to perfectly classify data in the provided test set. However, if achieving high performance is really necessary or if the performance needs to be improved, the group recommends trying other machine learning algorithms, such as a Random Forest or a Support Vector Machine, and compare their training time, cross-validation scores, and performance metrics to see which model will perform better for the given task. Improving the data could also be considered, as the dataset is only limited to a few out of the many physical characteristics a mushroom can exhibit.

References

1. Ministry of Forests. (n.d.). *What is a mushroom?* What is a Mushroom. https://www.for.gov.bc.ca/hfp/publications/00029/mushwhat.htm
2. Napoli, J. (2021, March 30). *Fungiculture: Everything you need to know about mushroom farming.* Feast and Field. https://feastandfield.net/read/fruits-and-vegetables/fungiculture-everything-you-need-to-know-about-mushroom-farming/article_a38ca72a-8e46-11eb-affb-db3ba58c0f54.html
3. MISSOURI DEPARTMENT OF CONSERVATION. (n.d.). *Destroying angel*. Missouri Department of Conservation. Retrieved April 20, 2023, from https://mdc.mo.gov/discover-nature/field-guide/destroying-angel
4. Huges, K. (2021, June 7). *Why are some mushrooms poisonous?* The Conversation. https://theconversation.com/why-are-some-mushrooms-poisonous-160151

5. Woodstream Corporation. (n.d.). *Growing mushrooms: How to deal with mushroom pests.* Safer® Brand Home, Garden & amp; Lawn. Retrieved April 20, 2023, from https://www.saferbrand.com/articles/mushroom-pests
6. UCI Machine Learning. (n.d.). *Mushroom classification.* Kaggle. https://www.kaggle.com/datasets/uciml/mushroom-classification?resource=download
7. Gurucharan, M. (2022, October 28). Gini index for decision trees: Mechanism, perfect & imperfect split with examples. upGrad blog.https://www.upgrad.com/blog/gini-index-for-decision-trees/
8. Javatpoint. (n.d.). Entropy in machine learning. www.javatpoint.com. https://www.javatpoint.com/entropy-in-machine-learning
9. Ye, A. (2020, September 9). When and why tree-based models (often) outperform neural networks. Medium. https://towardsdatascience.com/when-and-why-tree-based-models-often-outperform-neural-networks-ceba9ecd0fd8
10. Sharma, S. (2022, November 20). Activation functions in neural networks. Activation Functions in Neural Networks. https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

Appendix A. Contribution of Members

Table 6. Contribution of Members.

Name	Contributions
Josh Austin Mikhail T. Natividad	Performed methodology for the decision tree model, Contributed in writing chapters 1,3, 4, and abstract
Alexandra G. Padua	Contributed in writing chapter 4
Lloyd Dominic G. Refuerzo	Contributed in writing chapter 5
Joshua Emmanuel G. Tipon	Performed methodology for the neural network model, Contributed in writing chapters 3 and 4, formatted the paper
Christian Dave P. Tordillo	Contributed in writing chapter 2, formatted the paper