University of Tartu

Faculty of Science and Technology

Institute of Technology

Johanna Olesk

Mushroom genera determination using machine learning

Bachelor's Thesis (12 ECTS)

Curriculum Science and Technology

Supervisor:

Prof., PhD Gholamreza Anbarjafari

Tartu 2021

Abstract

Mushroom genera determination using machine learning

Mushroom determination using classification manuals is a tedious and time-consuming task for mycologists and mushroom hunters. Machine learning provides a tool to automate this process based on mushroom images using a small dataset. Since mushroom genera level classification has been understudied, it is important to direct attention to this matter. In this study, advanced machine learning algorithms were used in order to classify Cantharellus, Coprinus, Pholiota and Russula mushroom genera that are widely spread in Estonia. The classification was done based on the image grayscale pixels. To improve the classification accuracy, majority voting and mean rule methods from the ensemble-based classification were applied to the dataset. The highest accuracy obtained was 75.38%, with the majority voting method fusing five high performing classifiers. This study showed that ensemble methods improve the mushroom genera classification accuracy compared to individual classifiers. In addition to a novel approach of classifying mushrooms on the level of genera, a new labelled mushroom image dataset was collected that can be used in the future for similar studies.

CERCS:

T111 Imaging, image processing

Keywords:

Machine learning, Ensemble learning, Mushroom genera classification, Image classification

Kokkuvõte

Seente perekondade määramine masinõpet kasutades

Seente määramine klassikalisi määramismeetodeid kasutades on tülikas ja aeganõudev ülesanne nii mükoloogidele kui ka seenekorjajatele. Masinõppe abil on võimalik seente määramise protsess piltide põhjal automatiseerida, kasutades suhteliselt väikest seenepiltide andmekogumit. Kuna perekonna tasemel seente määramist on eelnevalt vähe uuritud, siis on tähtis sellele teemale tähelepanu pöörata. Eestis laialt levinud seeneperekondade Cantharellus, Coprinus, Pholiota ja Russula klassifitseerimiseks on kasutatud kõrgema taseme masinõppe algoritme. Klassifikatsioon on tehtud pildi halltooni pikslite põhjal. Ansambelõppe meetoditest kasutati häälteenamuse ja keskmise reegli võtet, et klassifitseerimistulemust parandada. Kõrgeim tulemus 75.38% saadi häälteenamuse võttega, kus kasutati viie algoritmi väljundit. Selle töö tulemused näitasid, et ansambelõppe meetodid parandavad seente perekondade klassifikatsiooni võrreldes individuaalsete masinõppe algoritmidega. Lisaks uudsele seene perekondade määramise käsitlusele koguti antud töö käigus ka uus sildistatud seenepiltide andmekogum, mida on võimalik tulevastes sarnastes töödes kasutusele võtta.

CERCS:
T111 Pilditehnika

Märksõnad:
Masinõpe, Ansambelõpe, Seente perekondade klassifitseerimine, Piltide klassifitseerimine

Contents

List of Figures

List of Tables

Abbreviations, definitions

AI - artificial intelligence

ANN - artificial neural network

CAD - computer aided diagnostic

CNN - convolutional neural network

FN - false negative

FP - false positive

GA - genetic algorithm

gcForest - multigrained cascade forest

HOG - histogram of oriented gradients

JPG - joint photographic group, lossy digital image compression format

KNN - k-nearest neighbour

MDAS - mushroom diagnosis assistance system

MLP - multi-layer perceptron

MR - magnetic resonance

PCA - principal component analysis

PNG - portable network graphics, lossless digital image compression format

RGB - red, green, blue colour channels in digital images

SROI - segmented regions of interest

SVM - support vector machine

TN - true negative

TP - true positive

Introduction

Mushrooms are fungi used by humans for food and medicinal purposes for hundreds of years due to their high nutrient and vitamin levels [1]. The species range of fungi is wide, estimated at around 2.2 to 3.8 million species [2] around the world and 5500 in Estonia [3], containing toxic as well as non-toxic mushroom species. Classification of mushrooms is usually done by mushroom classification manuals as paperback books or e-books. This, however, is a tedious and time-consuming way for real-time mushroom classification. In addition, as many mushroom species even from different genera can be similar, humans can be prone to misclassifying them, which may have serious health consequences when eaten.

Machine learning is an essential tool to learn a classification model [4] and has been researched for over 60 years [5], proving to be a valuable method for scientists to get insights from the data. Machine learning provides the means of automating the image-based classification of mushrooms on the level of toxicity, genera and species. Previously using machine learning to classify mushrooms on the level of toxicity has been mainly studied [6] [7]. However, classification based on toxicity is only beneficial for food purposes. Mycologists need to classify mushrooms on the level of species and genera. Nevertheless, only a few studies are done in order to develop a classification model to classify mushrooms based on images on the level of species [8] [9] and even fewer studies on the level of genera [8]. Those few works done, however, are based on deep learning, requiring big labelled datasets to work with. As large labelled mushroom datasets are difficult to find, there is a need for machine learning models that could obtain reliable results with smaller datasets. Considering these reasons, developing a method using machine learning to determine mushroom genera based on their images is an important task to tackle.

This thesis aims to utilise advanced machine learning methods in order to automate the classification of Cantharellus, Coprinus, Pholiota and Russula mushroom genera. These mushroom genera are widely spread in Estonia [10] [11] [12], and the automation of the determination of

these genera would benefit the mycologists in Estonia to classify mushrooms faster and easier. Due to the lack of data, no deep learning algorithms could be used in the current study. However, in order to improve the robustness and accuracy of the mushroom genera determination, several ensemble techniques, such as majority voting and mean rule, have been utilised.

The thesis is structured into five main parts. Chapter 1 gives an overview of the literature and is divided into three smaller sections. Section 1.1 explains machine learning and its applications in general, Section 1.2 provides an overview of the usage of machine learning in image classification, and Section 1.3 focuses on the previous work done on mushroom classification using machine learning. Chapter 2 describes the collected dataset. Chapter 3 outlines the methods used in the study and is divided into three part according to the experiments carried out. Chapter 4 brings out the results of each experimental phase which are further discussed in Chapter 5. The thesis ends with conclusions and future work.

1 Literature review

1.1 Machine learning

Human beings are able to observe, acquire new knowledge and skills but also modify the existing. In other words, they can learn by themselves. On the other hand, machines rely on data that is given to them; they can learn from past experience [5]. Machine learning is a core of artificial intelligence (AI), helping to overcome the bottlenecks of data acquisition [13] arising from the phenomenon of "Big Data" [4], where computing systems are able to gather and transport huge amounts of data. It enables the computers to learn from data, examples and previous experience, as well as modify their actions and improve the accuracy over time [5]. In machine learning, the goal is to learn a classification or prediction model. This way, machine learning proves to be a valuable tool for scientists to get insights, predictions and decisions from this big data [4].

The terms artificial intelligence and machine learning are known and researched for over 60 years [5]. The development of machine learning started with Alan Turing [14], who posed the question, "Can computers think?". In order to check the intelligence of the computer, he created a test called the Turing Test, where the machine should convince the human that they are talking to another human, not with a machine. Arthur Samuel [15] was the first to actually formulate the phrase "machine learning" in 1959, defining it as a field in which it should be easy to program computers to learn from experience, eventually eliminating the need for detailed programming effort completely. In his research, Samuel built a basic machine learning program to play checkers by looking ahead some moves and evaluating the positions on the board. He claimed that a computer could be programmed to learn to play checkers better than the person who created the program. A modern machine learning explanation by Tom Mitchell [16] from 1997 says that "A computer program is said to learn from experience E with respect to some class of tasks T, and performance measure P, if its performance at tasks in T, as measured by P,

improves with experience E".

Machine learning uses computational methods to recognise the patterns occurring in the data, make accurate predictions, and improve performance [17]. It uses past information that can be either human-labelled training sets or information obtained via interaction with the environment. The crucial factors in the final prediction accuracy are the quality and the size of the data [18].

The basis of machine learning is the design of efficient and accurate prediction algorithms with time, space and sample complexity. The success of the algorithm depends on the data used, making machine learning intertwined with statistics and data analysis. Therefore, machine learning combines fundamental concepts in computer science, statistics, probability and optimisation to learn from data iteratively and search for hidden patterns [18]. A diverse array of machine learning algorithms has been developed to solve different data and problem types, such as labelled or unlabelled data, speech recognition or computer vision tasks. As stated by Jordan and Mitchell [4], function approximation is the focus of many algorithms. In this case, the task is expressed in a function, and the accuracy is increased through learning. The function generally depends on the parameters and tunable degrees of freedom. During the model training, the best values are found to optimise the performance.

Machine learning can be divided into different categories depending on the data available, the way training data is received, and the test data used to evaluate the learning algorithm [18]. Most commonly, machine learning algorithms fall into three more significant categories [19]:

1. supervised learning,
2. unsupervised learning,
3. reinforcement learning.

This categorisation, however, is not the only way of classifying machine learning algorithms. Nowadays, the research has blended over these three categories [4], giving rise to other categories, such as active learning, semi-supervised learning, ensemble learning etc. In Figure 1.1, machine learning paradigms and their further approaches are presented.

1.1.1 Supervised machine learning

The most common approach in machine learning is supervised learning [20] [17] [4], also called predictive learning, which requires labelled data. It takes advantage of the function approxima-

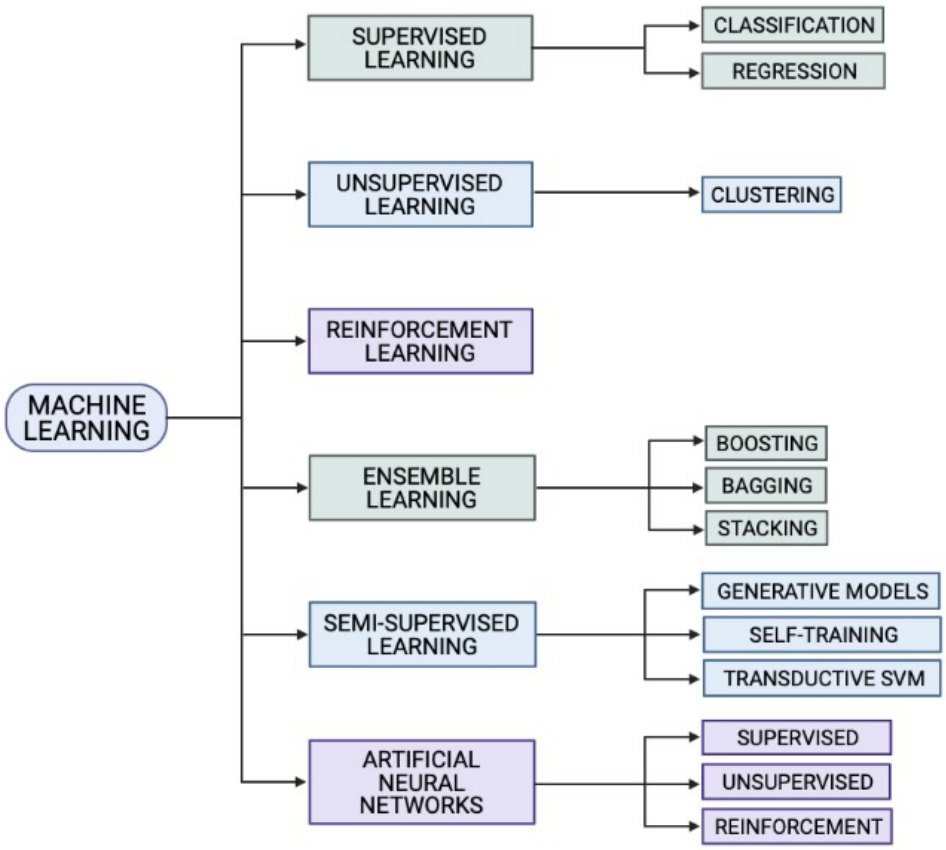


Figure 1.1: Machine learning paradigms and their further approaches.

tion in which the training set D consists of input-output pairs $D = \{(x_i, y_i)\}_{i=1}^{N})$, where N is the sample size or the number of training samples. The input x is called the feature, also named as the covariate or predictor. Features are often fixed dimensional vectors of numbers, for example, the pixels of an image. The output y is called the label, target or response. The predictions in supervised learning are formed via a learned mapping $f(x)$ from inputs $x \in \chi$ to outputs $y \in \gamma$. Supervised learning can be subcategorised as classification and regression [21], where classification has discrete labels while regression has continuous.

1.1.2 Unsupervised machine learning

Unsupervised learning [20] [17] [4] or descriptive learning algorithms do not need labelling of the data, avoiding the need for collecting large labelled datasets. In this case, only the inputs $D = \{(x_i\}_{i=1}^{N})$ are known and observed without corresponding outputs. Unsupervised learning allows the model to explain the inputs and find interesting patterns in the data. Clustering and dimension reduction are two categories of unsupervised learning.

1.1.3 Reinforcement learning

Reinforcement learning [20] [17] [4] is the third large class of machine learning. It is an intermediate type between supervised and unsupervised learning. Although the algorithm is not provided with the correct output, the training data indicates whether the output is correct or not. It is encoded by the policy $a = \pi(x)$, specifying the action in response to each possible input x. A reward signal is received in response to correct actions, and punishment signals in response to wrong actions. Compared to supervised and unsupervised learning, it is more challenging to make reinforcement learning work as the reward signal can only be given occasionally, therefore, resulting in a minimal amount of information.

1.1.4 Ensemble learning

Ensemble learning [22] [23] [24] is a state-of-the-art method in machine learning. It combines predictions of several individual machine learning algorithms of supervised and unsupervised learning in order to increase overall prediction accuracy. Each of the algorithms is trained on the same training data followed by the fusion of the output using one of the available methods, such as majority voting. Ensemble learning proves to be efficient in many ways. Firstly, it avoids overfitting as averaging different hypothesis made by several algorithms reduces the risk of choosing the wrong hypothesis. Secondly, by decreasing the risk of obtaining a local minimum and getting stuck there, it shows a great computational advantage. And last, a combination of different models extends the search space, and, therefore, a better fit is achieved.

1.1.5 Applications

Being a multi-disciplinary field, machine learning has many areas of applications in research, commerce as well as everyday life [5]. In biomedicine, machine learning has shown a great promise to revolutionise diagnostics and treatment by providing continuously adapted improved detection, diagnosis and treatment strategies [25]. McKinney *et al.* [26] presented an AI system that was able to increase the accuracy and efficiency of breast cancer screening compared to human experts. The system was based on image classification and trained to identify cancer. In the concept of smart cities, Alrashdi *et al.* [27] proposed an Anomaly Detection-IoT (Internet of Things) system based on a random forest machine learning algorithm. The system effectively detects IoT cyberattacks in a smart city. These are just a few examples of the endless

opportunities of utilising machine learning.

1.2 Machine learning in image classification

Classification belongs to the supervised learning category [5], where the output labels are discrete. It aims to predict a discrete class label from given input [20], in other words, to recognise and group objects into distinct categories. Classification is done by classifier algorithms that learn from the training data and assign new data to a particular class or category. The valid mapping function [28] $f : x \longrightarrow y$ is drawn, and class prediction is made by the classification model. The classification model is made more accurate with the help of features - parameters in the given task set.

The output is a set of L labels that are unordered and mutually exclusive [20], also known as classes $y = \{1, 2, ..., L\}$. There are four types of classification [4] (Figure 1.2). (1) Binary classification [17] has only two classes, meaning $L = 2$ and it is denoted by $y \in \{0, 1\}$. In case there are more than two labels, $L > 2$, the classification is (2) multi-class [17] where y takes one of L labels, denoted by $y \in \{0, 1, ..., L\}$. (3) Multi-label classification [29] occurs when y can simultaneously be labelled by several L labels. It is viewed as a prediction of multiple related binary class labels, also called multiple output model. Lastly, when classification is (4) imbalanced [30], the sample sizes in different classes vary significantly.

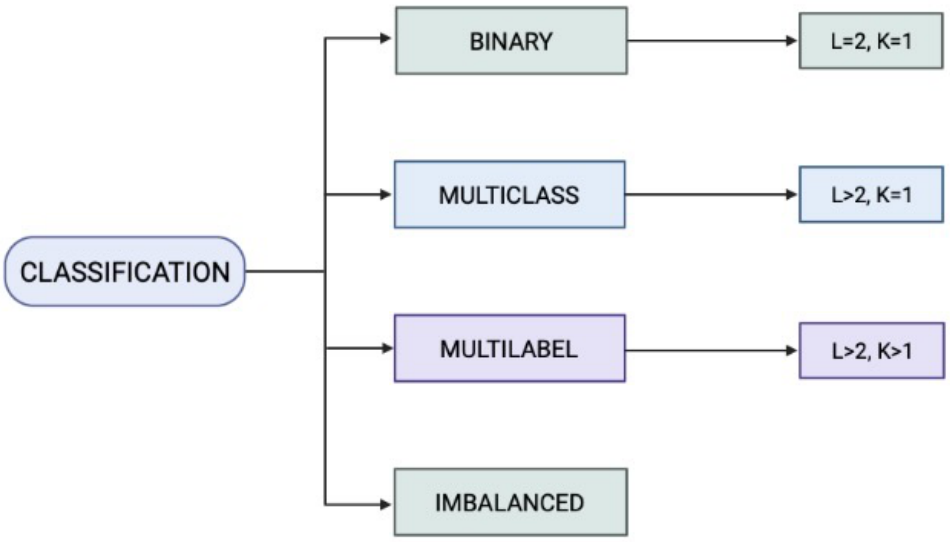


Figure 1.2: Types of classification. Here L corresponds to the number of labels and K corresponds to the number of labels one class may have.

Image classification [20] is a supervised learning task where the input set X is a set of images with high dimensionality. In the case of RGB images, there are three colour channels $C = 3$, red, green and blue, and $D_1 \times D_2$ pixels, resulting in an input $X = R^D$, where $D = C \times D_1 \times D_2$. Grayscale images have only one colour channel $C = 1$. Each pixel intensity is represented with an integer from range $\{0, 1, ..., 255\}$. Therefore, the image is represented by an array of numbers.

1.2.1 Workflow

The image classification model is built according to the general classification model steps. There are five main steps in creating an image classification model:

1. data collection,
2. data preparation and preprocessing,
3. feature selection,
4. classification algorithm selection and training,
5. evaluation.

Figure 1.3 visualises the overall classification workflow.

1.2.1.1 Data collection

Data collection consists of data acquisition, data labelling and improvement of existing data [31]. Techniques for data acquisition depend on the data wanted. Data discovery is used when new datasets are wanted. Data augmentation is complementing discovery by adding more external data to the existing datasets. Data generation creates synthetic or crowdsourced data when no external datasets are available. Data labelling can be done simultaneously with data acquisition or after data acquisition.

1.2.1.2 Data preparation and preprocessing

There can be many errors in the initial data, such as impossible, unlikely or missing values or irrelevant features [32]. Therefore, data preparation and preprocessing need to be carried out. Data preprocessing can be done either manually (in the case of smaller datasets) or with

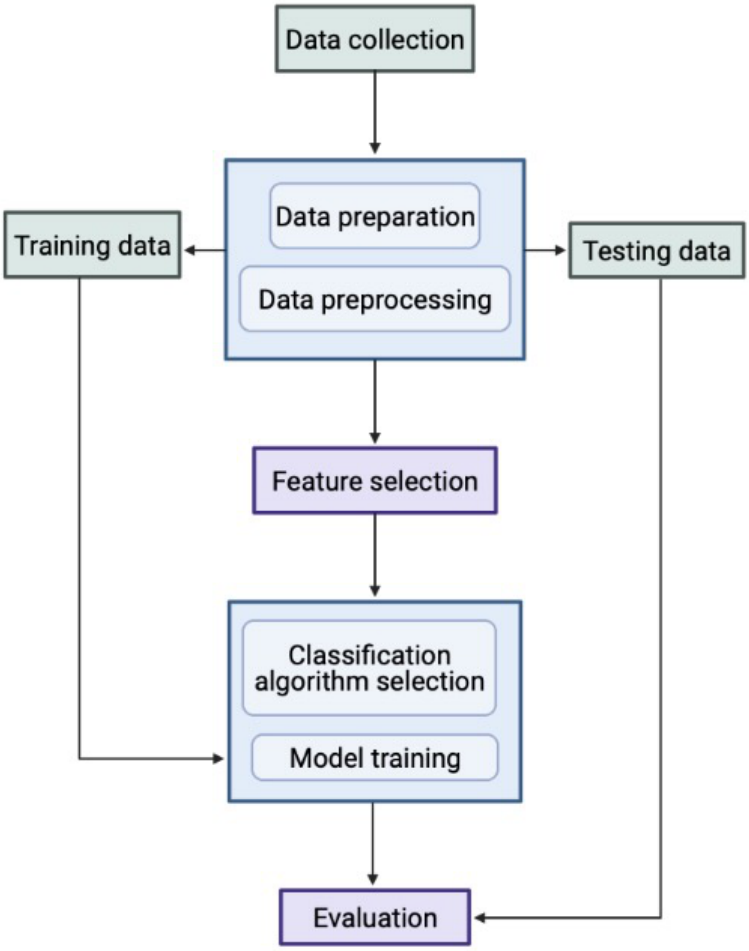


Figure 1.3: Image classification flowchart.

numerous methods. Missing data, for example, can be handled with methods introduced by Batista and Monard [33]. For outliers or noise detection, there are several techniques described by Hodge and Austin [34]. In the case of huge datasets, data sampling [32] may be carried out. Random sampling selects a subset of instances randomly from the whole dataset. If class values are distributed unevenly, other sampling techniques need to be used. Imbalanced data can cause models to favour over-represented class over other classes, causing misclassification [35]. Two methods for sampling imbalanced data are random undersampling (RUS) and random oversampling (ROS). In RUS, samples are randomly removed from the majority class, while in ROS, samples are randomly added to the minority class. In image classification, resizing and image processing are often carried out to bring out objects of interest or enhance the image.

1.2.1.3 Feature selection

In feature selection [32], relevant features are selected while irrelevant and redundant are removed in order to reduce the number of input variables. Feature selection algorithms have two parts where the selection algorithm generates a proposed subset of features and tries to find an optimal subset while the evaluation algorithm determines the performance. One of the commonly used techniques is principal component analysis (PCA). Relevant features for image classification include pixels, histogram of oriented gradients (HOG) or frequency domain, to name a few.

1.2.1.4 Classification algorithm and training

The critical step is to choose the classification algorithm [32] that performs the best as the evaluation step is usually based on the prediction accuracy. That depends largely on the input data and how many classes it has (is it a binary or multi-class classification problem). Then the training data is given into the classification algorithm in order to train the model [28]. Some of the most commonly used image classification algorithms are support vector machine, decision tree, random forest and logistic regression.

1.2.1.5 Evaluation

The evaluation step is important to get feedback on how well the trained classifier model performed and, according to that, optimise the parameters and choose the best performing model. For that test data set is given as input to the model, and the performance is calculated. There are different evaluation metrics [36] suitable for classification. A visually most detailed evaluation method is a confusion matrix that shows all correct and incorrect classifications for each class, rows corresponding to true labels and columns to predicted labels. From the confusion matrix, it is easy to calculate the prediction accuracies. Accuracy is the most popular method measuring the frequency at which the classifier makes the correct prediction and is calculated by the following formula:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{1.1}$$

where TP, TN, FP and FN are true positive, true negative, false positive and false negative, respectively. In the case of multi-class data, the following formula is used for calculating clas-

sification accuracy for each class:

$$class \quad accuracy = \frac{TP}{TP + FP} \tag{1.2}$$

Two useful metrics are precision and recall showing the quality of the model (i.e., how many positive class predictions actually belong to the positive class) and the quantity (i.e., the number of positive predictions out of all positive items), respectively. The formula for precision is

$$precision = \frac{TP}{TP + FP}, \tag{1.3}$$

while recall is calculated by the formula

$$recall = \frac{TP}{TP + FN} \tag{1.4}$$

Precision and recall metrics are often combined via their harmonic mean, also called the F1-score or f-score where

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall} \tag{1.5}$$

The model is considered perfect if the F1-score is one and failure if the F1-score is zero.

1.2.2 Classification algorithms

There are many well-studied algorithms for classification that are widely used for image classification. Which classification algorithm to use depends on the dataset it has to work on [37], i.e., whether it is a binary or multi-class classification problem. It is essential to compare the classifier accuracies and choose the best predictive one. Known algorithms include support vector machine, decision trees, random forest, k-nearest neighbours, naive Bayes, logistic regression and linear regression, to name a few.

1.2.2.1 Support Vector Machine

Support vector machine (SVM) algorithm [38] [39] [40] [41] [42] is a supervised learning algorithm proposed by Vapnik in 1995. This model is primarily used for classification by aiming to find an optimal hyperplane in order to segment the samples. The optimal hyperplane is the plane that correctly separates the classes while maximising the distance between them. SVM

can be divided into linear and non-linear models depending on whether the data domain can be linearly divided or not. The equation corresponding to the linear classification hyperplane for input data x is

$$w \cdot x + b = 0, \tag{1.6}$$

where w is the weight vector and b is the bias.

The optimal classification function is

$$f(x) = sgn[w^* \cdot x + b] = sgn[\sum_{i=1}^{n} a_i^* y_i (x_i \cdot x) + b^*], \tag{1.7}$$

where $sgn()$ is a sign function, w^* is optimal weight coefficient vector, n is the number of training samples, a^* is optimal Lagrange multiplier, y_i is the label value of sample i and b^* is optimal bias.

For linearly non-separable data, transformations need to be performed to map the original data into a linear classification problem in high-dimensional space by using kernel function $K(x_i, x_j)$, such as linear or radial based kernel function. Then the optimal hyperplane is found similarly to the linear SVM method, and the optimal classification function is

$$f(x) = sgn[\sum_{i=1}^{n} a_i^* y_i K(x_i, x) + b^*] \tag{1.8}$$

1.2.2.2 Decision Tree

A decision tree [28] [38] [43] is a logic-based supervised learning algorithm generating a set of decisions that will lead to the prediction of the class. The decision tree consists of nodes and branches. The node represents the feature in an instance that will be classified, and the branch represents the value of the node. The root node is the feature that best divides the training data. Although the decision tree method is simple, fast, easy to visualise and does not require much data preprocessing, it may result in very complex tree structures that are not generalised enough.

1.2.2.3 Logistic Regression

Logistic regression [20] [43] is another supervised learning algorithm to implement linear classification models. The core of the algorithm is a logistic function used to learn the parameters

of the model and predict the instances. The logistic function is

$$f(z) = \frac{1}{1 + e^{-z}}, \tag{1.9}$$

where $z := \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_n x_n$ and $x_1, x_2, ..., x_n$ are independent input variables. The dependent output variable $y = f(z)$. In case $L = 2$, we deal with binary logistic regression while in case $L > 2$ multinomial or multi-class logistic regression task.

1.2.2.4 Random Forest

Random forest algorithm [5] [20] [44] is a combination of decision trees. It is an ensemble learning technique. First, many decision trees are generated, and then the most popular class is chosen based on voting. It uses a bagging technique that involves training several classifiers resulting in ensemble output from the mean or majority voting of the decision trees.

1.2.3 Related work in image classification

Image classification is an important method in many fields, from biomedical imaging to robot sensing to the entertainment business and is actively used to solve problems in many research areas.

Sachdeva *et al.* [45] proposed a computer-aided diagnostic (CAD) system to segment and classify brain tumours from magnetic resonance (MR) images. Their system has four modules starting with the marking of tumour regions and saving them as segmented regions of interest (SROIs). The feature extraction is carried out in the second module, in which the intensity and texture features are extracted from SROIs. The third module consists of feature selection using a genetic algorithm (GA). Lastly, the selected features are given as inputs to the classification module that consists of SVM and artificial neural network (ANN), resulting in hybrid classifiers GA-SVM and GA-ANN using the standard multi-layer perceptron (MLP). The experiments were done on two brain tumour MR image datasets and compared with SVM and ANN classifier performances. Developed hybrid classifiers with GA showed higher prediction accuracy than that of classifiers without optimising GA. Also, the GA-ANN classifier showed greater overall accuracy on both datasets, 94% and 94.1%, respectively, while GA-SVM accuracy was 91.7% and 89%, respectively. The authors proposed using GA-SVM to find preliminary probability in identifying tumour class, and GA-ANN for accuracy confirmation.

In another study Shukla *et al.* [46] developed a framework to recognise six developmental

disorders from facial images of patients. This model can be used in order to make an initial diagnosis of these disorders. The model of Shukla *et al.* relies on a convolutional neural network (CNN) to extract the representations of the faces from the images. A machine learning algorithm linear SVM is then used for classification. They trained the SVM to classify the images as positive or negative classes according to the decision formed after training the feature vector obtained from the CNN. Their proposed method gave 98.8% accuracy, which exceeded other techniques.

In biological research fields, machine learning offers great opportunities for automating otherwise time-consuming, tedious work. Popescu and Sasu [47] presented machine learning based classification techniques for the field of palynology - a study of pollen and spores [48]. They investigated 12 classifiers, including naive Bayes, k-nearest neighbours (KNN) and random forest, on a public pollen dataset with a small number of images available. The traditional image classification workflow was followed where they first separated the pollen from the background and carried out image enhancement. They then extracted and selected the features, followed by training the classification models, tuning some of the hyperparameters based on trial and error. The authors compared all 12 classifiers, took naive Bayes as the baseline model, and found that the best performing models build predictions based on data clustered together. The best results were obtained by the KNN algorithm, receiving 83.43% classification accuracy.

Machine learning based image classification methodologies are utilised considerably in smartphone application development. Zhu and Spachos [49] built and evaluated traditional machine learning, deep learning and transfer learning methodologies to determine the optimal model for an Android application for butterfly classification. In the conventional machine learning field, they tested SVM and random forest, obtaining 52.5% and 72.3% accuracy, respectively. In comparison with deep learning and transfer learning methods which obtained the highest accuracies of 98.3% and 98.4%, respectively, SVM and random forest algorithms remained relatively simple. A recent study by Foysal *et al.* [50] was carried out to propose a smartphone application detecting costumer's body shape to provide them with optimal fitting clothing recommendation. Their proposed body shape detection algorithm combines image processing and machine learning to classify body shapes using KNN, giving an overall 87.5% accuracy.

As the examples showed, machine learning has proven to be an excellent tool for image classification tasks. Although the previously described accuracies of the models are relatively high, the classification model can be optimised, and the performance further improved. In

addition, the high-performing models were built on deep learning algorithms requiring large datasets. In the case of smaller datasets, machine learning models should be employed.

1.3 Mushroom classification

Mushrooms are fungi that have humans have consumed for food as well as medicinal purposes [1]. Mushrooms are low in energy but provide a high source of nutrients, vitamins and biologically active compounds believed to have an antitumour effect. The range of fungi species is broad, estimated at around 2.2 to 3.8 million [2] in the world, about 5500 in Estonia [3], in which there are edible as well as poisonous mushroom species. Lots of people go mushroom hunting by themselves, knowing little about mushrooms and how to differentiate them. Thus, the classification of mushrooms is vital to prevent accidental poisoning by eating the wrong type. Especially important would be developing an automatic mushroom classification application for smartphones to be used while mushroom hunting. There are previous studies done on mushroom classification, however, mainly on classifying whether the mushroom is edible or non-edible, only a few works on classifying mushroom genera or species. Also, most of the works done prior are based on the physical description of mushrooms rather than images. Therefore, developing machine learning models for mushroom genera and species classification based on images is an important task to tackle.

1.3.1 Mushroom classification based on images

Images provide several means of features to be extracted and used as input for the classification model. Maurya and Singh [6] proposed a machine learning based method for mushroom classification as edible or non-edible using texture features extracted from images. They started with preprocessing by resizing the mushroom images and converting them to grayscale. In the feature extraction step, colour and grayscale features are extracted to derive new features and reduce the dimensionality. For classification, the authors used SVM, KNN, decision tree, ensemble training and discriminant analysis to classify the mushrooms as edible or poisonous. Their dataset had 250 publicly available mushroom images. The authors concluded that the SVM classifier performed better with respect to other classifiers obtaining 76.6% accuracy. They claimed that the performance would be increased if the image background is removed in all images, so all extracted features would contain only mushroom features and not include background features.

Machine learning techniques to classify mushrooms as poisonous or non-poisonous were also studied by Ottom *et al.* [7]. They implemented several machine learning methods such as SVM, neural network, decision tree and KNN on mushroom images. After resizing the images, the authors extracted Eigenvalues with cap diameter, stem height and diameter, HOG and parametric features. They gave the features as inputs into the four machine learning classification models and compared the results. KNN showed the highest accuracy on real dimensions, reaching 94.4% precision on Eigen features. When real dimensions were replaced with virtual dimensions, SVM reached the highest accuracy of 87.6% on combined Eigen and parametric features. When the image background was removed, the best performing method was neural network on combined Eigen and histogram features with an accuracy of 84.1%.

Other studies on mushroom image classification have been done using neural network architectures which, however, require much bigger datasets for training. Hidalgo [8] presented a smartphone application MushroomApp to identify mushrooms from an image taken using an ANN classifier. In contrast to previous works described, Hidalgo's model classifies mushroom species from seven genera, not only if they are poisonous or not. The process occurs in two steps. First, the genus is predicted and then the species. The output of the ANN model is binary, meaning it will tell if it is the given class or not. For genus, the author tested the model with 27,000 and 10,000 images and, as expected, obtained higher performance with more images, resulting in an f-score of 0.68. For mushroom species, the model was tested with 7000 and 5000 images and four genera where again higher performance was obtained with more images, resulting in f-scores of 0.36, 0.47, 0.3 and 0.42. As a result of Hidalgo's work, a prototype of MushroomApp was developed. Sulc *et al.* [9] developed a fungi species recognition system based on deep convolutional neural networks. Deep learning, however, is out of the scope of the present study and will not be discussed further.

1.3.2 Mushroom classification based on physical description

There are more studies done on mushroom classification based on the description of mushroom physical attributes such as cap, odor, gills, stalk, veil, ring, spore, habitat, etc., mostly to classify if mushrooms are poisonous or non-poisonous. Based on those visual features Wang *et al.* [51], proposed a mushroom toxicity classification method using multigrained cascade forest (gcForest) and compared it with logistic regression and SVM classifiers. gcForest is a decision tree ensemble method that consists of multigrained scanning and cascade forest. gcForest resulted

in the highest average accuracy of 98.35% with fluctuation of less than 8%, therefore, proving to be the best classifier out of the three. However, the stability needs to be improved in the future, which will result in even higher accuracy.

Another ensemble-based mushroom classification method was proposed by Yildirim and Bingöl [52]. They employed and compared five different ensemble classification algorithms (subspace discriminant, RUSBoosted trees, subspace KNN, bagged trees and boosted trees) in order to predict if the mushroom is edible or not. The results showed that the best performing method was bagged trees resulting in 100% accuracy over four other classifiers.

Shaheed and Abd [53] developed a Mushroom Diagnosis Assistance System (MDAS) for smartphones in which two machine learning classifiers, naive Bayes and decision tree, are used to determine if the mushroom is edible or poisonous. They tested the classifiers without and with feature selection. The results showed that the decision tree performed better on both cases obtaining 98.96% and 99.99% accuracies, respectively, while naive Bayes showed results of 95.83% and 98.46%, respectively. Therefore, the authors concluded that feature selection improves the accuracy and, although the decision tree took a bit longer time to train, it resulted in higher accuracy.

As in image-based mushroom classification, also in physical description based classification, neural networks are a common approach. Alkronz *et al.* [54] proposed a multi-layer ANN to classify mushrooms as edible or poisonous. The architecture had one input layer, three hidden layers and one output layer, and it reached an accuracy of over 99%. As this is again out of the scope of current work, this will not be discussed further.

1.3.3 Importance of mushroom image classification

The works on mushroom classification brought out in the previous sections show that the studies have mainly focused on classifying mushrooms as poisonous or non-poisonous. However, this is not a sufficient level of classification for mycologists and people interested in the higher level of mushroom classification, such as genera or species level. Moreover, classifying mushrooms based on images is not as widely studied as image-based classification. However, images provide the classifier with visible features rather than merely a verbal description that can be misleading. Therefore, image-based classification can provide higher accuracy, especially if combined with a verbal description of the features not visual on the picture, such as distinct smell or colour of spores. Finally, although there are a few deep learning models done to

classify mushrooms on the species level, there are no studies on the level of genera. Therefore, developing machine learning classification models to classify mushrooms on the level of genera is important.

2 Dataset

2.1 Data description

The mushroom dataset (Figure 2.1) used was collected specifically for the current study. It consists of 1118 labelled images of ten mushroom genera: Agaricus, Amanita, Cantharellus, Coprinus, Cortinarius, Lactarius, Mycena, Pholiota, Pluteus and Russula. The classes are slightly imbalanced. The dataset includes images taken of mushrooms in the wild as well as after picking. There are images with and without background. The pictures of the mushrooms are taken either from the upper part of the cap, bottom part of the cap, profile or sectional cut. The final models use downsampled dataset consisting of five classes: mushroom genera Cantharellus, Coprinus, Pholiota, Russula, and merged class others (i.e., an equal amount of images from every other six genera).

Figure 2.1: A selection of images from the mushroom dataset. The images are resized to 200×200 pixels.

2.2 Data collection

Mushroom images were collected during July, August and September 2020 by Eduardo Fabian Garza Garza. A dataset with six genera and 700 images was collected and uploaded to the Google Drive folder, each genus into a different folder during July. More images were added during August, and the last 273 images were uploaded on September 3rd. The images were collected from three sources:

1. Mushrooms by Phillips [55],
2. Field guide to common macrofungi in eastern forests and their ecosystem functions [56],
3. MushroomExpert website [57].

All images in JPG format were converted to PNG format in order to keep the lossless compression and quality.

A two pixels wide rectangle was drawn around each mushroom on the image, one rectangle per image and one mushroom per rectangle, to localise the mushroom on the image. The rectangle separated the mushroom from the background of the image, making it easier for the classifier to recognise the pixels of the mushroom. The colour of the rectangle was chosen as unnatural as possible, in this case, neon green (with code 4CFF00, consisting 29.8% of red, 100% of green and 0% of the blue channel), to be separable from the image and mushroom itself.

Each mushroom genera was stored in a separate folder, which is considered a way of labelling. A table with information about the images was created (Table 2.1). The table contained the genera of the mushroom on the image (numbered from 0 to 9), the name of each image, which side the image was taken from (up, down, profile or sectional, numbered from 0 to 3), and whether the image had a background or not (numbered 1 or 0).

In addition, Eduardo Fabian Garza Garza compiled a table with the features of each mushroom genera. The table specifies the major group the genus belongs to. The cap of each genus is described in the means of its shape, surface, stickiness and colour. Hymenium type and gills are brought out, reporting the gills attachment, spacing and colour. The stalk shape and position, annulus and annulus colour are described together with veil, veil type and colour. The substrate (soil or wood) and lactation of the mushrooms were indicated, and whether the genus

is gregarious or not. This table was later used to identify the distinct features of the best and the worst classified genera.

Each genus was assigned an abbreviation for the use in confusion matrices. The abbreviations are as follows: C1, C2, C3, C4, C5, C6, C7, C8, C9 and C10, corresponding to Agaricus, Amanita, Cantharellus, Coprinus, Cortinarius, Lactarius, Mycena, Pholiota, Pluteus and Russula, respectively. These abbreviations will be used in figures and tables throughout the study.

Table 2.1: An example of the created dataset table with selected samples. In the column "Genera", the numbers 0-9 correspond to genera Agaricus, Amanita, Cantharellus, Coprinus, Cortinarius, Lactarius, Mycena, Pholiota, Pluteus and Russula, respectively. The name of the image file is in the column "Image name". Column "Mushroom side" indicates whether the image is taken from the upper part of the cap, bottom part of the cap, profile or sectional cut, numbered 0-3, respectively. The "Background" column shows if the image has a background or a uniform colour behind the mushroom, 0 indicating no background and 1 the presence of background.

Genera	**Image name**	**Mushroom side**	**Background**
0	Agaricus4-3.png	1	0
0	Agaricus32agaricus_auricolor_01-1.png	0	1
1	Amanita1-1.png	2	0
1	Amanita6-5.png	1	0
2	Cantharellus4-4.png	0	0
2	Chantarellus19cantharellus_cf_cibarius_01-1.png	2	1
3	Coprinus1-1.png	3	0
3	Coprinus14-3.png	0	1
4	Cortinarius2-3-1.png	2	0
4	Cortinarius19-2.png	1	0
5	Lactarius26-1.png	2	1
5	Lactarius55-1.png	0	1
6	Mycena7-2.png	3	0
6	Mycena32-5-2.png	2	0
7	Pholiota1-1.png	1	0
7	Pholiota32pholiota_limonella_03-3.png	2	1
8	Pluteus7-1.png	1	0
8	Pluteus53pluteus_longistriatus_03-1.png	0	1
9	Russula48-3.png	1	0
9	Russula63-1.png	2	0

2.3 Data preparation and feature selection

Only profile images (Figure 2.2) from the dataset were chosen for further testing for the current study since this simplified the classification process. For that, the dataset was sorted, and profile

picture images were extracted. As the initial dataset was slightly imbalanced (i.e., each class had a slightly different number of samples), data sampling needed to be carried out. From each class, 65 random samples were chosen. Then the dataset was split into initial training and testing datasets where 80% of the images were reserved for training (52 images from each class) and 20% for testing (13 images from each class). These training and testing datasets were used throughout the study.

The images in the training and testing datasets were preprocessed. Each image was resized to 200×200 pixels and converted to grayscale (i.e., the image has only one colour channel instead of three R, G and B channels) (Figure 2.3). The images were flattened into an array containing pixels that were selected as features and input to the classification models. The labels of each corresponding image were saved in another array.

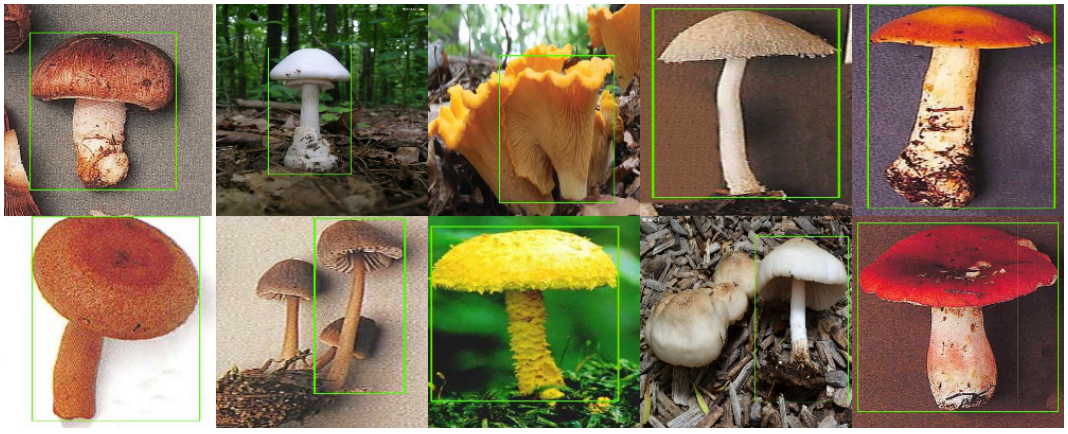


Figure 2.2: Examples of the chosen profile mushroom images. Images are resized to 200×200 pixels.

Figure 2.3: Mushroom images after the conversion to grayscale. Images are resized to 200×200 pixels.

2.4 Data downsampling

In the later phases of the classification, only four genera out of all ten genera (here and after genus is interchangeably used with class) were utilised. That required downsampling of the original training and testing datasets. Four chosen genera (Cantharellus, Coprinus, Pholiota and Russula) were extracted from the datasets, and the other six genera were merged to form one class called "others". As the new datasets were very imbalanced where the class called "others" was highly dominating, data downsampling needed to be carried out in order to balance all the classes. Cantharellus, Coprinus, Pholiota and Russula genera each contained 52 samples in the training set and 13 samples in the testing dataset. Class "others" had 6×52 samples in the training and 6×13 samples in the testing dataset. From every six genera present in the previously mentioned class, an equal number of samples were randomly selected in order to have 52 samples from the class "others" in total in the training dataset and 13 samples in the testing dataset. The new downsampled training dataset contains 5×52 samples, and the testing dataset 5×13 samples.

3 Methods

3.1 Adopted classification algorithms

A list of 17 machine learning multi-class classification algorithms was chosen (Table 3.1) according to their compatibility to classify multi-class datasets. The grayscale pixel features and labels of the prepared training dataset of ten classes were converted into Pandas dataframe and given as an input to each of the classifiers. The testing dataset features and labels were converted to dataframe the same way. The feature dataframe was given as input to the trained models to obtain the predicted labels and classification accuracy. Predicted classes of each classifier were saved in a list. Confusion matrices of each classifier were saved on an Excel file, and the accuracy of each class prediction was calculated according to Equation 1.2. The columns on the confusion matrix tables represent the predicted classes,labelled orange, while the rows represent the true classes, marked yellow. The accuracies for each class are indicated on the last column of each table, labelled green, and total accuracy represents the accuracy of the overall classification model, i.e., the average of the class accuracies is calculated to obtain the model accuracy. The confusion matrix indicates how many samples the classifier classified as true positives (TP) and how many it classified as false positives (FP).

For each class, the average class accuracy amongst all classifiers was calculated. This was done separately for each class by summing up the class accuracies provided by all 17 classifiers and divided by the number of classifiers (i.e., 17). From the average class accuracies, the best-classified mushroom genera were chosen, and data downsampling was carried out (described in more detail in Section 2.4). The best-classified genera were Cantharellus, Coprinus, Pholiota and Russula, which were chosen to be tested further and be the basis of this study. Other mushroom genera were merged to a class named ”others” and downsampled to 65 samples where each genus had an equal number of samples presented. The same number of samples was maintained for each mushroom genera (65 samples) as in the previous experiment, the

same 80% of it going for the training set and 20% for the testing set.

Table 3.1: Multi-class classification algorithms chosen for this study and their description. Classifiers marked in bold were selected for ensemble-based classification in the later phases of the study.

Algorithm	**Description**
Decision tree	More details in Subsection 1.2.2.2.
Extra tree classifier	Extremely randomised decision tree classifier, should be used within ensemble methods.
Extra trees classifier	It uses averaging to improve the predictive accuracy of extra tree classifiers and reduce over-fitting.
Gaussian Naive Bayes	Assumes the likelihood of the features to follow Gaussian distribution and is based on Gaussian Naive Bayed algorithm $P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2})$, where x is the feature, μ_y is the mean of values in x and σ_y^2 the Bessel corrected variance.
KNN	Finding a predefined number of training samples closest to the sample to be predicted and predict the label based on these samples. The value of k is specified by user and stays constant, in the scope of this study $k = 3$.
Label propagation	Semi-supervised learning algorithm where a small subset of samples have labels which are propagated to the unlabelled points. A KNN kernel was used with $k = 3$.
Label spreading	Semi-supervised learning algorithm similar to label propagation, however, it is more robust to the noise present in the data. In this study a KNN kernel with $k = 3$ was used.
Linear discriminant analysis	Linear supervised classification algorithm using Bayes' rule to find a linear combination of features characterising/separating the classes.
LinearSVC	Implementation of SVM (Subsection 1.2.2.1) with linear kernel. Hinge loss and multi-class strategy crammer singer optimising a joint objective over all classes were used in this study.
Logistic regression	More details in Subsection 1.2.2.3. In this study multi-class option was set to multinomial.
Logistic regression with CV	Implementation of logistic regression using K-fold cross validation to get improved accuracy. The number of folds used in this study was $cv = 5$.
MLP classifier	A neural network classifier having input layer, output layer and hidden layers inbetween. The architecture of MLP classifier for the first experimental phase had three hidden layers with 100, 100 and 150 nodes, and for the second experimental phase 200, 150 and 100 nodes, respectively.
Nearest centroid	Similar to KNN classifier. Here every class is represented by a centroid and a class of a sample is predicted based on the nearest centroid.
Quadratic discriminant analysis	A variant of linear discriminant analysis allowing non-linear classification of data.
Random forest	More details in Subsection 1.2.2.4. Maximum depth of the trees were chosen to be 15.
Ridge classifier	Classifier converting target values to -1 and +1 to treat them as regression task.
Ridge classifier with CV	Implementation of Ridge classifier using K-fold cross validation for improved accuracy.

3.2 Classification of Cantharellus, Coprinus, Pholiota and Russula genera

After choosing the best-classified mushroom genera (Cantharellus, Coprinus, Pholiota and Russula) out of all genera, the previous classification steps were repeated with the new downsampled training and testing datasets. Five classes instead of ten classes were classified (other mushrooms, Cantharellus, Coprinus, Pholiota and Russula). The same 17 classifiers were used (Tabel 3.1), and trained classification models were saved for further testing. The predicted labels of the testing dataset were saved in a list. Confusion matrices, described in Section 3.1, were saved on an Excel file, and performance accuracy of each class prediction was calculated according to Equation 1.2. The overall model accuracy was obtained, taking an average of class accuracies of one model. From the obtained results, five high performing classifiers (linearSVC, decision tree, extra tree classifier, logistic regression and random forest) were chosen for the classification by ensemble learning (Table 3.1 classifiers marked in bold).

3.3 Ensemble based decision making

For ensemble learning based classification, five high accuracy classifiers (linearSVC, decision tree, extra tree classifier, logistic regression and random forest) were chosen. Downsampled testing dataset was used where each sample was given to each previously saved classifier at the same time one by one. The pixel features were converted into Pandas dataframe and given as input to each of the classifiers. The results were saved in a list, each classifier having a separate list for predictions.

3.3.1 Majority voting

The first ensemble technique tested was majority voting using five classifiers. The majority voting method [58] takes the predictions of each classifier and considers each of them as a vote. The majority of the votes will declare the final prediction for the ensemble technique. The decision is described by the following equation:

$$\sum_{t=1}^{T} d_{t,J} = max_{j=1}^{C} \sum_{t=1}^{T} d_{t,j}, \tag{3.1}$$

where d is the decision of the t^{th} classifier, $t = 1, \ldots, T$ and $j = 1, \ldots, C$, where T is the number of classifiers and C is the number of classes. The probability of the success of the ensemble method is found by the equation

$$P_{ens} = \sum_{k=(T/2)+1}^{T} \binom{T}{k} p^k (1-p)^{T-k}, \tag{3.2}$$

where each classifier has a success rate of p.

The results of mushroom classification using linearSVC, decision tree, extra tree, logistic regression and random forest classifiers were combined by majority voting (Figure 3.1.). For every sample, the prediction of each classifier was checked. The number of times that the classifiers predicted each class was counted, and the class obtaining the majority of the votes was declared as the final class. The predictions were visualised on a confusion matrix, described in Section 3.1. The accuracy of each class prediction was calculated according to Equation 1.2. The overall model accuracy was obtained, taking an average of class accuracies of one model. In order to show the improvement introduced by the majority voting technique using five classifiers, an average of all class accuracies of the five classifiers was taken. The averaged class accuracies were compared to the corresponding class accuracies of the majority voting technique. The same comparison was made for the overall ensemble model accuracy and the average of the classifiers model accuracies.

3.3.2 Mean rule

Another ensemble technique next to the majority voting based ensemble method used was the mean rule [58], a simple algebraic combiner with a normalisation factor. In order to obtain the final prediction, the average of all classifier j^{th} outputs is calculated using the formula

$$\mu_j(x) = \frac{1}{T} \sum_{t=1}^{T} d_{t,j}(x), \tag{3.3}$$

where $\mu_j(x)$ is the final class of the sample.

In the case of the mean rule, however, only classifiers providing confidence score can be used. A confidence score is a value from zero to one showing the probability of the sample falling into each class. The higher the probability, the higher the confidence. In this study linearSVC, logistic regression and random forest classifiers could be used as these classifiers provided confidence scores. The decision tree and extra tree classifier only provided binary

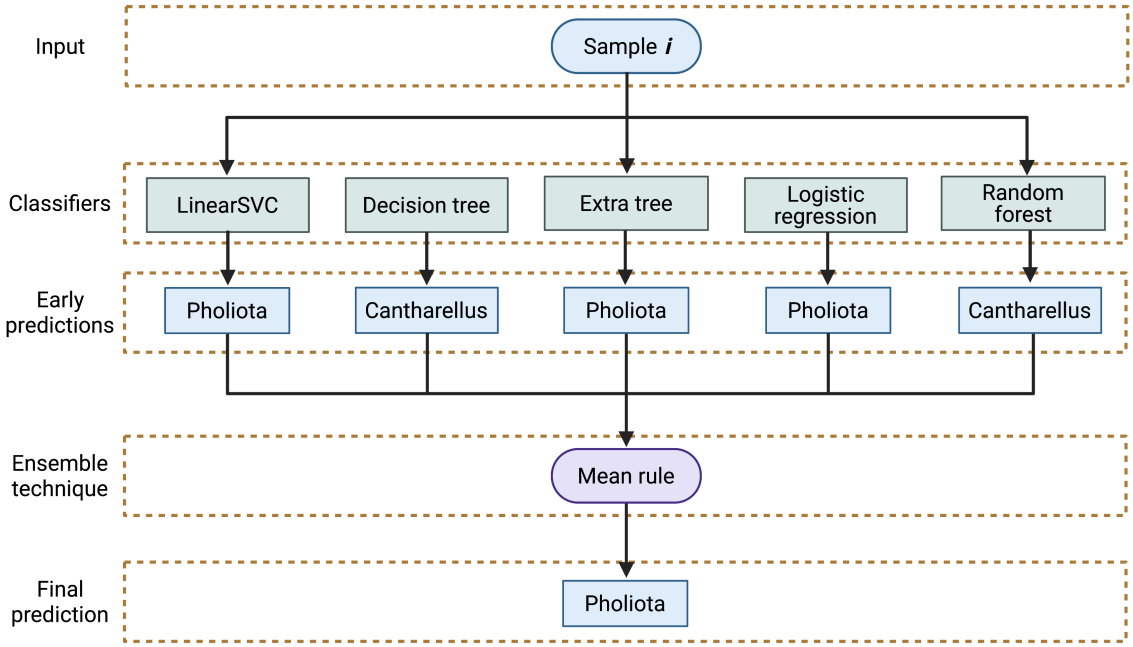


Figure 3.1: Majority voting. The workflow of the majority voting technique presented by sample i. The predictions on sample i from each classifier are combined by majority voting, and the majority of the votes declares the final class of the ensemble method.

results, i.e., one for the predicted class and zero for all the other classes.

The random forest provided confidence scores for each sample in a list, while the outputs for linearSVC and logistic regression were distances to which the predicted sample lies from the hyperplane, also saved in a list for each sample. From those distances, confidence scores can be calculated for each sample by using the formula

$$confidence \quad score = \frac{l - min(l)}{\Sigma l}, \tag{3.4}$$

where each distance l in the list is subtracted by the minimum distance l in the list and then divided by the sum of all distances l. The obtained confidence values were saved in a new list for the mean rule ensemble technique.

The mean rule formula (Equation 3.3) was applied to linearSVC, logistic regression and random forest classifier outputs (Figure 3.2). From the obtained list of new confidence values, the maximum value was declared as the predicted class. A confusion matrix, described in Section 3.1, was compiled, and each class prediction accuracy was calculated according to Equation 1.2. The overall model accuracy was obtained, taking an average of class accuracies of one model. In order to show the improvement introduced by the mean rule technique fusing three classifiers, an average of all class accuracies of the three classifiers was taken. The av-

eraged class accuracies were compared to the corresponding class accuracies of the mean rule technique. The same comparison was made for the overall ensemble model accuracy and the average of the classifiers model accuracies.

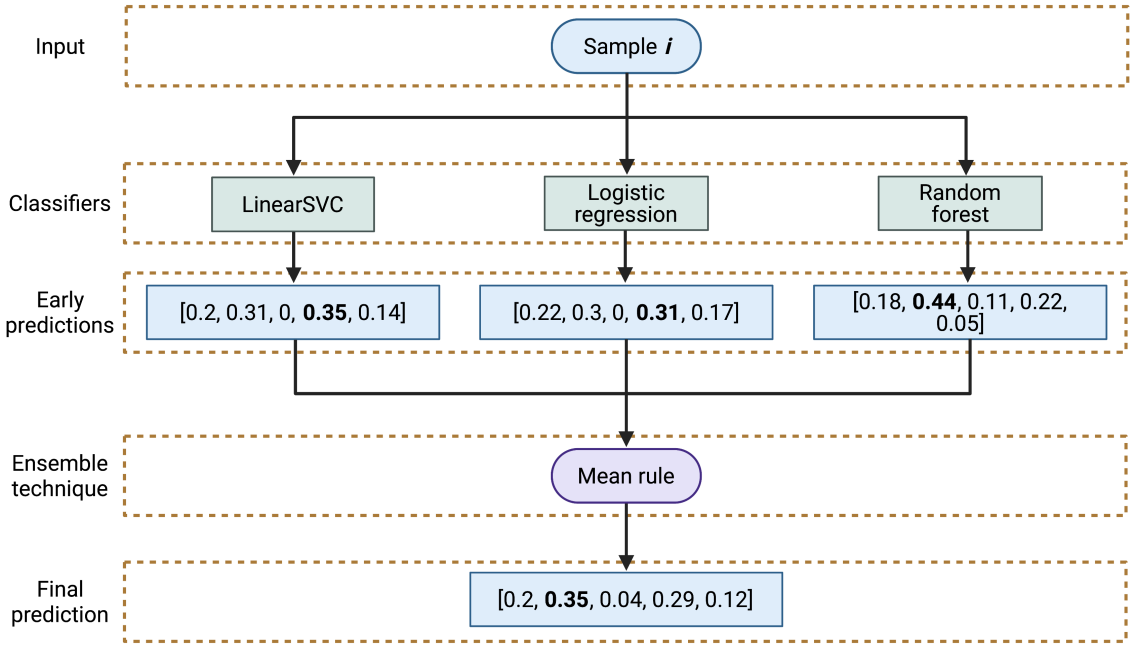


Figure 3.2: Mean rule. The workflow of the mean rule technique presented by sample i. The mean rule equation is applied to the outputs in the form of confidence scores of all classifiers. The class with the highest calculated confidence score (marked in bold in the final prediction section) is the predicted class of the ensemble classifier.

3.3.3 Majority voting using linearSVC, logistic regression and random forest

The majority voting method, described in Subsection 3.3.1, was applied to the three classifiers used in the mean rule based ensemble method. This was carried out in order to test which method, majority voting or mean rule, will give better results when the same number of classifiers is used. Also, testing the majority voting method fusing three classifiers gave the possibility to compare this method against the same ensemble method fusing five classifiers in order to learn which one shows higher results.

The workflow followed was the same as in Subsection 3.3.1, where classification results of each classifier (linearSVC, logistic regression and random forest) were checked, and the frequency of each class prediction occurrence was counted. The class with the majority of the votes was declared as the final class. The results were visualised on a confusion matrix,

described in Section 3.1, and the accuracy of each class prediction was calculated according to Equation 1.2. The overall model accuracy was obtained, taking an average of class accuracies of one model. In order to show the improvement introduced by the majority voting technique using three classifiers, an average of all class accuracies of the three classifiers was taken. The averaged class accuracies were compared to the corresponding class accuracies of the majority voting technique. The same comparison was made for the overall ensemble model accuracy and the average of the classifiers model accuracies.

4 Experimental results

4.1 Classification of all genera

The results of the first experiment where all 17 classifiers were trained on all ten classes of the grayscale mushroom dataset are presented in Figure 4.1. From the results, the highest classified genera were chosen. This was done by taking an average of the class accuracies for each class (Table 4.1). Four genera with the highest average accuracies were selected. The best results were obtained for the Cantharellus genus, where the average accuracy obtained was 67.87%, followed by Coprinus, Russula and Pholiota, with average classification accuracies of 55.66%, 54.33% and 51.13%, respectively. The worst classified genera were Amanita, Pluteus and Lactarius with classification accuracies of 17.65%, 24.89% and 25.21%, respectively.

As for the classification model results, the accuracies are displayed in Table 4.2. The highest performing classifier was random forest, reaching 53.08% classification accuracy. Four other classifiers (extra trees, logistic regression, logistic regression with CV and linearSVC) obtained accuracy higher than 50%, while all other classification models had prediction accuracy staying under 45%.

Table 4.1: The average class accuracies.

Class	Accuracy (%)
Cantharellus	**67.87**
Coprinus	**55.66**
Russula	**54.33**
Pholiota	**51.13**
Mycena	49.77
Cortinarius	30.32
Agaricus	28.96
Lactarius	25.21
Pluteus	24.89
Amanita	17.65

Linear SVC

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Accuracy (%)
C1	2	2	1	0	2	1	1	1	0	3	15.38
C2	1	4	2	0	2	0	0	1	3	0	30.77
C3	0	1	10	0	0	1	0	1	0	0	76.92
C4	0	0	0	9	1	0	1	0	0	2	69.23
C5	1	1	0	0	5	3	1	0	1	1	38.46
C6	1	1	1	1	3	4	0	0	1	1	30.77
C7	1	1	1	0	1	1	7	1	0	0	53.85
C8	3	0	1	0	0	0	0	9	0	0	69.23
C9	0	1	2	1	0	0	0	1	6	2	46.15
C10	0	0	0	1	2	0	0	0	0	10	76.92
Total accuracy											50.77

Decision Tree

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Accuracy (%)
C1	4	1	1	2	1	2	0	1	0	1	30.77
C2	1	5	0	2	1	0	2	0	2	0	38.46
C3	0	1	8	1	0	1	0	1	1	0	61.54
C4	1	0	1	8	0	0	0	0	1	2	61.54
C5	0	0	2	2	2	3	0	1	1	2	15.38
C6	1	0	1	2	1	2	1	1	2	2	15.38
C7	2	0	3	2	3	1	2	0	0	0	15.38
C8	1	0	3	0	0	0	1	7	1	0	53.85
C9	0	3	2	1	0	2	0	0	5	0	38.46
C10	0	1	2	1	1	2	1	0	0	5	38.46
Total accuracy											36.92

Extra Tree Classifier

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Accuracy (%)
C1	4	0	2	2	1	1	0	1	2	0	30.77
C2	3	2	1	2	0	1	1	2	1	0	15.38
C3	1	0	8	0	0	1	1	0	1	1	61.54
C4	0	1	0	8	1	0	1	0	1	1	61.54
C5	2	0	0	0	6	2	1	1	0	1	46.15
C6	1	3	0	0	1	1	3	0	1	3	7.69
C7	1	1	0	0	1	1	9	0	0	0	69.23
C8	2	0	1	1	0	0	0	8	1	0	61.54
C9	0	2	5	0	1	1	0	0	3	0	23.08
C10	2	0	2	0	3	3	0	0	1	4	30.77
Total accuracy											40.77

Extra Trees Classifier

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Accuracy (%)
C1	5	1	1	0	2	0	1	0	0	3	38.46
C2	1	4	2	0	3	0	1	0	2	0	30.77
C3	0	1	11	0	0	0	0	1	0	0	84.62
C4	0	0	0	9	1	0	1	0	0	2	69.23
C5	1	0	0	0	5	4	2	0	0	1	38.46
C6	1	0	2	0	2	4	1	0	1	2	30.77
C7	0	0	1	2	0	3	7	0	0	0	53.85
C8	1	0	1	0	1	0	0	10	0	0	76.92
C9	1	1	5	0	0	2	0	0	3	1	23.08
C10	0	0	1	0	0	2	0	0	0	10	76.92
Total accuracy											52.31

Gaussian Naive Bayes

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Accuracy (%)
C1	6	1	1	1	2	1	0	0	1	0	46.15
C2	2	2	4	1	1	0	1	0	1	1	15.38
C3	0	4	6	0	2	0	1	0	0	0	46.15
C4	0	1	0	7	3	1	0	0	0	1	53.85
C5	0	0	0	0	9	1	3	0	0	0	69.23
C6	1	0	1	1	4	3	2	1	0	0	23.08
C7	0	0	2	1	5	1	4	0	0	0	30.77
C8	2	0	2	2	1	0	4	2	0	0	15.38
C9	1	1	6	0	1	1	0	1	2	0	15.38
C10	0	0	1	1	5	2	1	0	0	3	23.08
Total accuracy											33.85

KNN

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Accuracy (%)
C1	6	1	1	0	1	2	1	0	0	1	46.15
C2	4	2	4	0	2	0	1	0	0	0	15.38
C3	2	0	10	1	0	0	0	0	0	0	76.92
C4	1	0	3	7	0	1	1	0	0	0	53.85
C5	1	0	3	0	4	3	2	0	0	0	30.77
C6	2	0	5	0	1	4	0	0	0	1	30.77
C7	0	0	4	0	0	1	8	0	0	0	61.54
C8	2	1	3	1	1	4	0	2	0	0	15.38
C9	4	1	5	0	0	2	0	0	1	0	7.69
C10	3	0	1	0	0	2	1	0	0	6	46.15
Total accuracy											38.46

Label Propagation

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Accuracy (%)
C1	6	1	1	0	1	2	1	0	0	1	46.15
C2	4	2	4	0	2	0	1	0	0	0	15.38
C3	2	0	10	1	0	0	0	0	0	0	76.92
C4	1	0	3	7	0	1	1	0	0	0	53.85
C5	1	0	3	0	4	3	2	0	0	0	30.77
C6	2	0	5	0	1	4	0	0	0	1	30.77
C7	0	0	4	0	0	1	8	0	0	0	61.54
C8	2	1	2	1	1	4	0	2	0	0	15.38
C9	4	1	5	0	0	2	0	0	1	0	7.69
C10	3	0	1	0	0	2	1	0	0	6	46.15
Total accuracy											38.46

Label Spreading

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Accuracy (%)
C1	3	0	1	0	0	3	2	0	1	3	23.08
C2	3	0	5	0	2	1	1	0	0	1	0.00
C3	0	1	10	1	0	1	0	0	0	0	76.92
C4	0	0	1	6	1	3	2	0	0	0	46.15
C5	0	0	2	0	2	5	4	0	0	0	15.38
C6	1	0	0	0	1	6	3	0	0	2	46.15
C7	0	0	1	0	0	1	11	0	0	0	84.62
C8	0	0	4	0	1	5	1	2	0	0	15.38
C9	2	0	8	0	0	2	0	0	1	0	7.69
C10	0	0	0	1	0	3	2	0	0	7	53.85
Total accuracy											36.92

Linear Discriminant Analysis

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Accuracy (%)
C1	3	3	0	1	4	0	1	0	0	1	23.08
C2	1	2	1	1	4	1	2	0	1	0	15.38
C3	0	1	9	0	0	1	1	0	1	0	69.23
C4	0	2	1	6	0	2	1	1	0	0	46.15
C5	0	3	0	0	3	2	2	1	1	1	23.08
C6	0	1	0	0	5	2	1	1	2	1	15.38
C7	0	0	0	0	2	3	7	0	1	0	53.85
C8	0	1	0	0	1	1	0	10	0	0	76.92
C9	1	0	2	1	1	1	0	1	4	2	30.77
C10	0	1	0	0	3	1	1	0	0	7	53.85
Total accuracy											40.77

Logistic Regression

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Accuracy (%)
C1	4	2	0	0	2	2	1	1	0	1	30.77
C2	2	3	2	0	3	0	1	0	2	0	23.08
C3	0	2	9	0	0	1	0	1	0	0	69.23
C4	0	0	0	9	1	0	1	0	0	2	69.23
C5	1	1	0	0	5	4	1	0	1	0	38.46
C6	1	1	1	1	4	4	0	0	1	0	30.77
C7	0	1	0	0	0	1	9	0	2	0	69.23
C8	2	2	0	0	0	0	0	9	0	0	69.23
C9	2	0	1	0	1	1	0	0	6	2	46.15
C10	0	0	0	1	2	1	0	0	0	9	69.23
Total accuracy											51.54

Logistic Regression with CV

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Accuracy (%)
C1	4	2	0	0	2	1	1	1	1	1	30.77
C2	3	3	2	1	2	0	0	0	2	0	23.08
C3	0	2	9	0	0	1	0	1	0	0	69.23
C4	0	0	0	9	1	0	1	0	0	2	69.23
C5	0	2	0	0	5	4	1	0	1	0	38.46
C6	1	1	1	1	4	4	0	0	1	0	30.77
C7	0	1	0	0	0	1	9	0	2	0	69.23
C8	2	2	0	0	0	0	0	9	0	0	69.23
C9	2	0	1	0	0	1	0	0	6	2	46.15
C10	0	0	0	1	0	2	1	0	0	9	69.23
Total accuracy											51.54

MLP Classifier

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Accuracy (%)
C1	0	2	2	0	1	0	0	5	0	3	0.00
C2	0	3	1	0	0	0	2	4	0	3	23.08
C3	0	2	4	0	0	0	0	6	0	1	30.77
C4	0	3	3	4	0	0	0	2	0	1	30.77
C5	0	2	2	0	1	0	2	4	0	2	7.69
C6	0	3	2	0	1	0	1	4	0	2	0.00
C7	0	0	4	0	0	0	1	8	0	0	7.69
C8	0	2	2	0	0	0	0	9	0	0	69.23
C9	0	1	5	0	2	0	0	1	0	4	0.00
C10	0	1	0	0	0	0	0	2	0	10	76.92
Total accuracy											24.62

Nearest Centroid

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Accuracy (%)
C1	3	4	1	2	1	0	0	0	0	2	23.08
C2	2	1	6	1	2	0	1	0	0	0	7.69
C3	0	0	9	0	0	0	3	1	0	0	69.23
C4	1	1	1	5	1	0	1	1	0	2	38.46
C5	0	0	0	0	5	3	3	0	1	1	38.46
C6	1	0	1	1	3	3	1	1	1	1	23.08
C7	0	0	2	0	0	4	5	0	2	0	38.46
C8	2	0	2	2	0	0	4	3	0	0	23.08
C9	2	0	7	0	0	0	1	0	2	1	15.38
C10	0	0	1	1	0	0	1	0	2	8	61.54
Total accuracy											33.85

Quadratic Discriminant Analysis

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Accuracy (%)
C1	1	0	4	0	3	1	2	1	0	1	7.69
C2	0	2	2	0	2	2	1	1	1	2	15.38
C3	2	0	6	2	1	0	0	1	0	1	46.15
C4	1	1	1	5	0	0	2	1	1	1	38.46
C5	3	2	0	0	0	1	2	2	1	2	0.00
C6	0	1	2	1	2	2	3	0	1	1	15.38
C7	4	0	0	1	0	1	3	1	1	2	23.08
C8	1	0	0	0	3	2	1	5	1	0	38.46
C9	0	3	0	0	1	1	0	0	1	7	7.69
C10	0	3	2	1	1	1	0	1	1	3	23.08
Total accuracy											21.54

Random Forest

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Accuracy (%)
C1	7	1	0	0	0	1	3	0	0	1	53.85
C2	2	2	4	1	2	0	0	0	1	1	15.38
C3	0	1	11	0	0	0	0	1	0	0	84.62
C4	1	0	0	8	0	0	2	0	0	2	61.54
C5	1	0	0	0	3	7	2	0	0	0	23.08
C6	1	0	2	0	1	6	0	0	1	2	46.15
C7	0	0	2	2	1	0	7	0	1	0	53.85
C8	1	2	0	0	0	0	0	9	1	0	69.23
C9	0	0	3	0	1	1	0	1	6	1	46.15
C10	0	0	0	0	0	1	1	0	1	10	76.92
Total accuracy											53.08

Ridge Classifier

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Accuracy (%)
C1	3	1	1	0	1	1	2	1	0	3	23.08
C2	2	1	3	0	1	1	0	1	3	1	7.69
C3	0	0	10	0	0	1	0	1	1	0	76.92
C4	0	1	0	8	2	1	1	0	0	0	61.54
C5	2	1	0	0	4	1	1	0	2	2	30.77
C6	2	0	0	0	3	5	0	0	1	2	38.46
C7	1	1	1	0	2	1	6	0	0	1	46.15
C8	3	0	0	1	0	0	0	9	0	0	69.23
C9	0	0	4	0	0	0	0	3	4	2	30.77
C10	0	1	1	0	1	0	0	1	1	8	61.54
Total accuracy											44.62

Ridge Classifier with CV

Class	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Accuracy (%)
C1	3	1	1	0	1	1	2	1	0	3	23.08
C2	2	1	3	0	1	1	0	1	3	1	7.69
C3	0	0	10	0	0	1	0	1	1	0	76.92
C4	0	1	0	8	2	1	1	0	0	0	61.54
C5	2	1	0	0	4	1	1	0	2	2	30.77
C6	2	0	0	0	3	5	0	0	1	2	38.46
C7	1	1	1	0	2	1	6	0	0	1	46.15
C8	3	0	0	1	0	0	0	9	0	0	69.23
C9	0	0	4	0	0	0	0	3	4	2	30.77
C10	0	1	1	0	1	0	0	1	1	8	61.54
Total accuracy											44.62

Figure 4.1: Confusion matrices of classifiers on the whole dataset. C1, C2, C3, C4, C5, C6, C7, C8, C9 and C10 correspond to mushroom genera Agaricus, Amanita, Cantharellus, Coprinus, Cortinarius, Lactarius, Mycena, Pholiota, Pluteus and Russula, respectively. The columns (orange) represent the predicted classes, the rows (yellow) true classes and the accuracy column (green) the accuracy of each class. The intensity of the red colour helps to visualise the results of the confusion matrices.

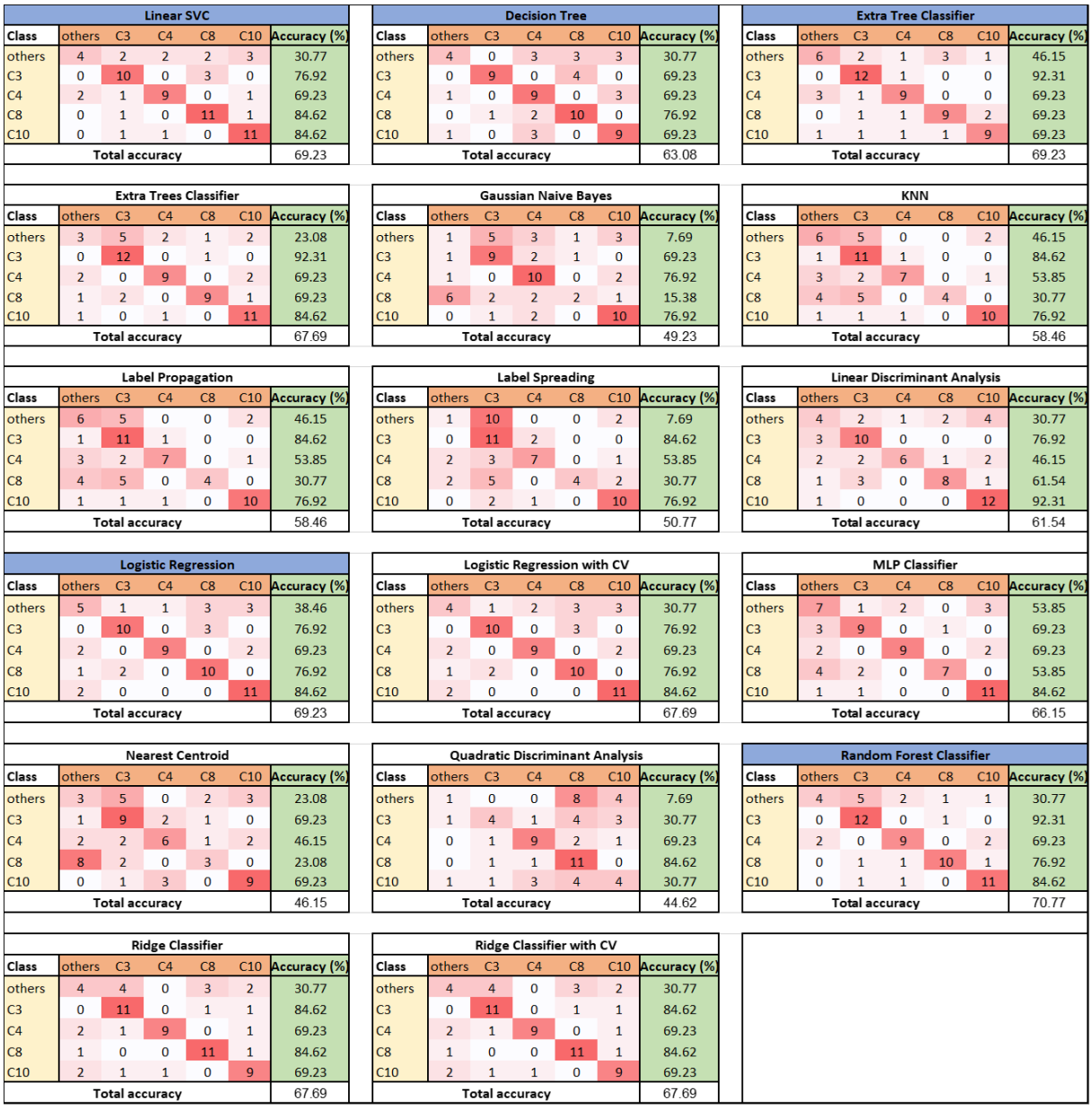
Table 4.2: Classifier accuracies for dataset with ten classes.

Classifier	Accuracy (%)
Random forest	53.08
Extra trees classifier	52.31
Logistic regression	51.54
Logistic regression with CV	51.54
LinearSVC	50.77
Ridge classifier	44.62
Ridge classifier with CV	44.62
Extra tree classifier	40.77
Linear discriminant analysis	40.77
KNN	38.46
Label propagation	38.46
Decision tree	36.92
Label spreading	36.92
Gaussian Naive Bayes	33.85
Nearest centroid	33.85
MLP classifier	24.62
Quadratic discriminant analysis	21.52

4.2 Classification of Cantharellus, Coprinus, Pholiota and Russula genera

The results of the second experiment phase (Figure 4.2) were obtained by training 17 classifiers on the downsampled dataset with five classes. Downsampled dataset contained the four genera chosen from the results of the previous experimental phase (Cantharellus, Coprinus, Pholiota and Russula) and a class class called ”others” where the other six genera were merged. The confusion matrix description is brought out in Section 4.1.

The highest performing trained models were random forest, linearSVC, extra tree and logistic regression with accuracies of 70.77%, 69.23%, 69.23% and 69.23% (Table 4.3), respectively. From the results of this experiment, classifiers for ensemble-based decision making were chosen. Four highest performing classifiers were included. Also, a decision tree model with the accuracy of 63.08% as a fifth classifier was added as for the majority voting based ensemble technique used in the later step, it was beneficial to have an odd number of classifiers. A decision tree classifier was chosen because it is a known and widely used algorithm which in this study also produced good results. The selected classifiers for ensemble-based classification are marked with a blue caption in Figure 4.2. Each classifier trained on the downsampled dataset



Linear SVC

Class	others	C3	C4	C8	C10	Accuracy (%)
others	4	2	2	2	3	30.77
C3	0	10	0	3	0	76.92
C4	2	1	9	0	1	69.23
C8	0	1	0	11	1	84.62
C10	0	1	1	0	11	84.62
Total accuracy						69.23

Decision Tree

Class	others	C3	C4	C8	C10	Accuracy (%)
others	4	0	3	3	3	30.77
C3	0	9	0	4	0	69.23
C4	1	0	9	0	3	69.23
C8	0	1	2	10	0	76.92
C10	1	0	3	0	9	69.23
Total accuracy						63.08

Extra Tree Classifier

Class	others	C3	C4	C8	C10	Accuracy (%)
others	6	2	1	3	1	46.15
C3	0	12	1	0	0	92.31
C4	3	1	9	0	0	69.23
C8	0	1	1	9	2	69.23
C10	1	1	1	1	9	69.23
Total accuracy						69.23

Extra Trees Classifier

Class	others	C3	C4	C8	C10	Accuracy (%)
others	3	5	2	1	2	23.08
C3	0	12	0	1	0	92.31
C4	2	0	9	0	2	69.23
C8	1	2	0	9	1	69.23
C10	1	0	1	0	11	84.62
Total accuracy						67.69

Gaussian Naive Bayes

Class	others	C3	C4	C8	C10	Accuracy (%)
others	1	5	3	1	3	7.69
C3	1	9	2	1	0	69.23
C4	1	0	10	0	2	76.92
C8	6	2	2	2	1	15.38
C10	0	1	2	0	10	76.92
Total accuracy						49.23

KNN

Class	others	C3	C4	C8	C10	Accuracy (%)
others	6	5	0	0	2	46.15
C3	1	11	1	0	0	84.62
C4	3	2	7	0	1	53.85
C8	4	5	0	4	0	30.77
C10	1	1	1	0	10	76.92
Total accuracy						58.46

Label Propagation

Class	others	C3	C4	C8	C10	Accuracy (%)
others	6	5	0	0	2	46.15
C3	1	11	1	0	0	84.62
C4	3	2	7	0	1	53.85
C8	4	5	0	4	0	30.77
C10	1	1	1	0	10	76.92
Total accuracy						58.46

Label Spreading

Class	others	C3	C4	C8	C10	Accuracy (%)
others	1	10	0	0	2	7.69
C3	0	11	2	0	0	84.62
C4	2	3	7	0	1	53.85
C8	2	5	0	4	2	30.77
C10	0	2	1	0	10	76.92
Total accuracy						50.77

Linear Discriminant Analysis

Class	others	C3	C4	C8	C10	Accuracy (%)
others	4	2	1	2	4	30.77
C3	3	10	0	0	0	76.92
C4	2	2	6	1	2	46.15
C8	1	3	0	8	1	61.54
C10	1	0	0	0	12	92.31
Total accuracy						61.54

Logistic Regression

Class	others	C3	C4	C8	C10	Accuracy (%)
others	5	1	1	3	3	38.46
C3	0	10	0	3	0	76.92
C4	2	0	9	0	2	69.23
C8	1	2	0	10	0	76.92
C10	2	0	0	0	11	84.62
Total accuracy						69.23

Logistic Regression with CV

Class	others	C3	C4	C8	C10	Accuracy (%)
others	4	1	2	3	3	30.77
C3	0	10	0	3	0	76.92
C4	2	0	9	0	2	69.23
C8	1	2	0	10	0	76.92
C10	2	0	0	0	11	84.62
Total accuracy						67.69

MLP Classifier

Class	others	C3	C4	C8	C10	Accuracy (%)
others	7	1	2	0	3	53.85
C3	3	9	0	1	0	69.23
C4	2	0	9	0	2	69.23
C8	4	2	0	7	0	53.85
C10	1	1	0	0	11	84.62
Total accuracy						66.15

Nearest Centroid

Class	others	C3	C4	C8	C10	Accuracy (%)
others	3	5	0	2	3	23.08
C3	1	9	2	1	0	69.23
C4	2	2	6	1	2	46.15
C8	8	2	0	3	0	23.08
C10	0	1	3	0	9	69.23
Total accuracy						46.15

Quadratic Discriminant Analysis

Class	others	C3	C4	C8	C10	Accuracy (%)
others	1	0	0	8	4	7.69
C3	1	4	1	4	3	30.77
C4	0	1	9	2	1	69.23
C8	0	1	1	11	0	84.62
C10	1	1	3	4	4	30.77
Total accuracy						44.62

Random Forest Classifier

Class	others	C3	C4	C8	C10	Accuracy (%)
others	4	5	2	1	1	30.77
C3	0	12	0	1	0	92.31
C4	2	0	9	0	2	69.23
C8	0	1	1	10	1	76.92
C10	0	1	1	0	11	84.62
Total accuracy						70.77

Ridge Classifier

Class	others	C3	C4	C8	C10	Accuracy (%)
others	4	4	0	3	2	30.77
C3	0	11	0	1	1	84.62
C4	2	1	9	0	1	69.23
C8	1	0	0	11	1	84.62
C10	2	1	1	0	9	69.23
Total accuracy						67.69

Ridge Classifier with CV

Class	others	C3	C4	C8	C10	Accuracy (%)
others	4	4	0	3	2	30.77
C3	0	11	0	1	1	84.62
C4	2	1	9	0	1	69.23
C8	1	0	0	11	1	84.62
C10	2	1	1	0	9	69.23
Total accuracy						67.69

Figure 4.2: Confusion matrices of classifiers on the downsampled dataset. C3, C8 and C10 correspond to mushroom classes Cantharellus, Coprinus, Pholiota and Russula, respectively. ”others” class consists of a mix of all other mushroom genera. Classifiers chosen for ensemble classification are marked with blue caption. The columns (orange) represent the predicted classes, the rows (yellow) true classes and the accuracy column (green) the accuracy of each class. The intensity of the red colour helps to visualise the results of the confusion matrices.

has introduced an increase in classification accuracy. The greatest increase occurred with the MLP classification model that showed 41.53%. Extra tree classifier increased 28.46% compared to training on the whole dataset and decision tree 26.16%. The increase in other classification models was between 12.30-23.08%.

Table 4.3: Classifier accuracies for dataset with five classes.

Classifier	Accuracy (%)
Random forest	**70.77**
Extra tree classifier	**69.23**
LinearSVC	**69.23**
Logistic regression	**69.23**
Extra trees classifier	67.69
Logistic regression with CV	67.69
Ridge classifier	67.69
Ridge classifier with CV	67.69
MLP classifier	66.15
Decision tree	**63.08**
Linear discriminant analysis	61.54
KNN	58.46
Label propagation	58.46
Label spreading	50.77
Gaussian Naive Bayes	49.23
Nearest centroid	46.15
Quadratic discriminant analysis	44.62

4.3 Ensemble based decision making

The results of the majority voting method using five classifiers, mean rule using three classifiers and majority voting using three classifiers are indicated in Table 4.4, Table 4.5 and Table 4.6, respectively. The specifics of the confusion matrices are described in detail in Section 4.1. The results of the chosen classifiers for ensemble-based classification were obtained by training them on the downsampled dataset of five mushroom classes (Cantharellus, Coprinus, Pholiota, Russula and others). The results for majority voting using five classifiers were obtained by combining the output of chosen five classifiers (linearSVC, decision tree, extra tree, logistic regression, and random forest) according to Subsection 3.3.1. The mean rule based ensemble method results were obtained by fusing the output of three classifiers (linearSVC, logistic regression and random forest) described in Subsection 3.3.2. Lastly, majority voting results with three classifiers were obtained by fusing the output of three classification models (linearSVC, logistic regression and random forest) according to Subsection 3.3.3.

The majority voting technique based ensemble classification yielded the highest, 75.38% accuracy. The best-classified genus was Cantharellus, obtaining 92.31% prediction accuracy, followed by Russula 84.62%, Pholiota 76.92% and Coprinus 69.23% accuracies. The lowest accuracy was observed in the merged class “others”, with a prediction accuracy of 53.85%.

The mean rule based method using three classification models achieved an overall accuracy of 73.85%, where the genera Cantharellus, Pholiota and Russula obtained an equal prediction accuracy of 84.62%, followed by Coprinus 69.23%. Also, in this method, the lowest accuracy was observed in the class ”others”, reaching 46.15%. Lastly, the majority voting method fusing the same three classifiers as for the mean rule method (linearSVC, logistic regression and random forest) yielded the lowest classification model accuracy, reaching only 69.23%. That is 4.62% lower than the mean rule based ensemble classification model and 6.15% lower than the majority voting based ensemble model using five classifiers. The best-classified genus was Pholiota 84.62% followed by Cantharellus and Russula, both with prediction accuracy of 76.92% and Coprinus 69.23%. The lowest prediction accuracy was again observed on the class ”others” with 38.46% accuracy.

Table 4.7 shows the change in the accuracy of each class compared to the average accuracy and the ensemble model accuracy change compared to the average of the individual classification model accuracies fused for the ensemble technique. Significant increases in the classes were observed in the majority voting technique using five classifiers. The highest rise in accuracy was observed in class ”others” in the majority voting method using five classifiers, giving an increase of 18.47%. The mean rule based method obtained an increase of 12.82% and majority voting using three classifiers 5.13%.

Table 4.4: Confusion matrices of majority voting method fusing linearSVC, decision tree, extra tree, logistic regression and random forest. C3, C8 and C10 correspond to mushroom classes Cantharellus, Coprinus, Pholiota and Russula, respectively. ”others” class consists of a mix of all other mushroom genera. The columns (orange) represent the predicted classes, the rows (yellow) true classes and the accuracy column (green) the accuracy of each class. The intensity of the red colour helps to visualise the results of the confusion matrices.

Class	others	C3	C4	C8	C10	**Accuracy (%)**
others	7	1	0	2	3	53.85
C3	0	12	0	1	0	92.31
C4	2	0	9	0	2	69.23
C8	0	1	1	10	1	76.92
C10	1	0	1	0	11	84.62
Total						75.38

Table 4.5: Confusion matrices of mean rule method fusing linearSVC, logistic regression and random forest. C3, C8 and C10 correspond to mushroom classes Cantharellus, Coprinus, Pholiota and Russula, respectively. ”others” class consists of a mix of all other mushroom genera. The columns (orange) represent the predicted classes, the rows (yellow) true classes and the accuracy column (green) the accuracy of each class. The intensity of the red colour helps to visualise the results of the confusion matrices.

Class	others	C3	C4	C8	C10	**Accuracy (%)**
others	6	2	0	2	3	46.15
C3	0	11	0	2	0	84.62
C4	2	0	9	0	2	69.23
C8	0	1	0	11	1	84.62
C10	1	0	1	0	11	84.62
Total						73.85

Table 4.6: Confusion matrices of majority voting method fusing linearSVC, logistic regression and random forest. C3, C8 and C10 correspond to mushroom classes Cantharellus, Coprinus, Pholiota and Russula, respectively. ”others” class consists of a mix of all other mushroom genera. The columns (orange) represent the predicted classes, the rows (yellow) true classes and the accuracy column (green) the accuracy of each class. The intensity of the red colour helps to visualise the results of the confusion matrices.

Class	others	C3	C4	C8	C10	**Accuracy (%)**
others	5	2	1	2	3	38.46
C3	0	10	0	3	0	76.92
C4	2	0	9	0	2	69.23
C8	0	1	0	11	1	84.62
C10	1	1	1	0	10	76.92
Total						69.23

Table 4.7: The improvement introduced by ensemble techniques. The accuracies of each class in the ensemble techniques and the ensemble model accuracy were compared to the average of the corresponding class accuracy of corresponding classifiers and the average of the individual classification model accuracies.

Class	**Majority voting (5 classifiers) accuracy difference (%)**	**Mean rule (3 classifiers) accuracy difference (%)**	**Majority voting (3 classifiers) accuracy difference (%)**
Others	18.47	12.82	5.13
Cantharellus	10.77	2.57	-5.13
Coprinus	0	0	0
Pholiota	0	5.13	5.13
Russula	6.16	0	-7.70
Model	7.07	4.11	-0.51

5 Discussion

As indicated in Section 4.1, the best-classified genera chosen for further classification in this study were Cantharellus, Coprinus, Pholiota and Russula, with average accuracies 67.87%, 55.66%, 54.33% and 51.13%, respectively. The prediction accuracy of the Cantharellus genus was over 12% better than the accuracies of the following three best genera. Cantharellus genus (Figure 5.1) has some very distinct features making the classification of the mushrooms belonging to this genus easier [59]. Cantharellus is the only genus out of ten genera with funnel or trumpet-shaped cap and decurrent gills (extending downward and partially wrapping the stalk) that are usually also visible on the image, explaining the significantly better classification accuracy. Moreover, the species within the Cantharellus genus are visually very similar, which also benefits the classification at the level of genera.

(a) First example of a species from Cantharellus genus mushroom.

(b) Second example of a species from Cantharellus genus mushroom.

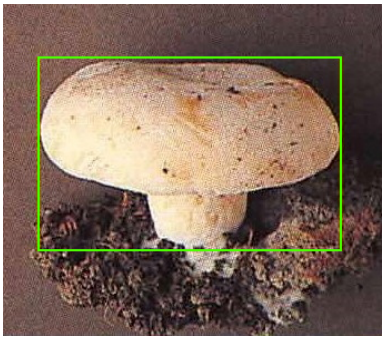
Figure 5.1: Cantharellus genus species examples.

The worst classified genera are Amanita, Pluteus and Lactarius, with classification accuracies of 17.65%, 24.89% and 25.21%, respectively. Amanita genus images in the mushroom dataset have several very different looking species [60], differing by colour and features such as visible gills (if they are also visible on the image), flaky cap, the shape of the cap and so on. An example of images of the Amanita genus but that of different species are shown in Figure 5.2. Also, in the case of Lactarius and Pluteus genera, the dataset has images of different species of Lactarius [61] and Pluteus [62] with distinctly different features such as the colour, the shape of the cap and whether the gills are visible on the image. Figure 5.3 shows two different species of Lactarius genus, while in Figure 5.4, two species of Pluteus genus are shown. High misclassification of these genera might be caused by species exhibiting very different features within each genus. In addition, in this study, the Pluteus genus is mostly misclassified with the Cantharellus genus. The reason might be because some species within the genus are similar to the Cantharellus genus, taken into account the colour and the cap shape.

(a) First example of a species from Amanita genus mushroom.

(b) Second example of a species from Amanita genus mushroom.

Figure 5.2: Amanita genus species examples.



(a) First example of a species from Lactarius genus mushroom.

(b) Second example of a species from Lactarius genus mushroom.

Figure 5.3: Lactarius genus species examples.

(a) First example of a species from Pluteus genus mushroom.

(b) Second example of a species from Pluteus genus mushroom.

Figure 5.4: Pluteus genus species examples.

The results of the second experiment brought out in Section 4.2 show that downsampling the dataset from ten mushroom classes to five classes (Cantharellus, Coprinus, Pholiota, Russula and merged class "others") significantly increases the prediction accuracy of each model. Downsampling the dataset to fewer classes decreases the complexity of the computations done by the classification model [63], hence leading to higher accuracy in prediction. The algorithms performing the best are one of the most known and commonly used ones in machine learning tasks (Table 4.3), justifying these algorithms to be widely used.

The results of ensemble-based classification brought out in Section 4.3 indicate that by fusing different high performing classifiers, the classification results are higher than that of individual classifier results. The accuracies of the individual classifiers linearSVC, decision tree, extra tree classifier, logistic regression and random forest are 69.23%, 63.08%, 69.23%, 69.23% and 70.77%, respectively (classifiers marked with blue caption in Figure 4.2). As seen from Table 4.4, combining these five classifiers with the majority voting method improves the classification accuracy, giving the final accuracy 75.38%. That is 4.61% higher than that of the best performing individual classification model random forest and 7.07% higher than the average of the accuracies of all individual classifiers, proving that this ensemble method does improve the prediction accuracy.

The mean rule based ensemble learning method fusing linearSVC, logistic regression and random forest algorithms gives an accuracy of 73.85% (Table 4.5). That is significantly higher than the accuracies of any of the three classifiers alone (linearSVC, logistic regression and random forest classifiers marked with blue caption on Figure 4.2). Although the accuracy is slightly lower (1.53%) than in the case of the majority voting based method fusing five classifiers, the computational cost of using three classifiers is cheaper. Additionally, the excluded classifiers, namely, decision tree and extra tree classifiers, have several downsides: they tend to overfit the data, and the calculations get very complex [64]. The mean rule method combining three classifier outputs results in slightly lower accuracy than the majority voting method combining five classifier outputs as more classifiers will provide the ensemble technique with more values to fuse.

The majority voting technique combining three classifiers used for mean rule based ensemble classification, however, results in lower accuracy than the mean rule based classification as it is a simpler method of ensemble learning. The prediction accuracy stays around the same level of each three classifier performances individually, reaching 69.23% (Table 4.6). Although using three classifiers is better, it can be deduced that it must be taken into account the method

of ensemble technique to compensate for the fewer number of classifiers. Furthermore, it can be said that the mean rule based method performs better compared to the majority voting technique, achieving higher accuracy when the same number of classifiers are used. The use of mean rule based classification has grown over time due to its simplicity and wide availability of applications [58].

Overall, the results of Section 4.3 indicate that ensemble techniques improve the classification accuracy as they reduce the variance of the model compared to individual models. There are several reasons behind it [22]. First, fusing three or five classifiers allow reducing the error made by each classifier as the error is compensated by the other classification models. Second, ensembling extends the search space; therefore, better fitting the data space, which is positive if the optimal class lies outside the space of any individual classification model.

Conclusion and future work

The aim of this study was to classify mushroom genera Cantharellus, Coprinus, Pholiota and Russula using different machine learning algorithms and techniques. Several individual machine learning classifiers were utilised, and the best performing models were fused in ensemble-based classification in order to learn if this technique improves the classification accuracy. The results of individual classifiers and different ensemble classification techniques were compared and analysed.

The results show that the best-classified genera are Cantharellus, Coprinus, Pholiota and Russula. Due to its distinct features from all other genera, Cantharellus achieves the highest classification accuracy amongst all mushroom classes. By downsampling the dataset of ten mushroom genera to five classes (Cantharellus, Coprinus, Pholiota, Russula genera and merged class others consisting of other six genera), the prediction accuracies of each classification model as well as each class are improved significantly. This is consistent with the idea that the fewer classes the dataset has, the better classification accuracy is achieved.

Throughout the experiments, it is shown that the best performing classifiers are the most known and commonly used algorithms, such as random forest, logistic regression, extra tree and linearSVC (based on SVM), proving that the usage of widely known algorithms is reasonable. By applying ensemble-based techniques and fusing the outputs of individual classifiers, the performance of classification is improved, supporting the idea of ensemble learning to produce better results than individual classification models. Also, when comparing different ensemble techniques, such as majority voting and mean rule, when the majority voting method fuses more classifiers then the mean rule method, it achieves higher results. However, when the same number of classifiers is used for both methods, the mean rule based method performs better than majority voting. These results are consistent with the suggestions from previous researchers to use the mean rule ensemble classification, as in addition to its good results, it also provides simplicity and is widely applicable.

It can be concluded that the majority voting based method using five classifiers improves the classification accuracy the most due to the use of more classifiers. However, the mean rule based method still obtains higher results if the same number of classifiers is used compared to majority voting. In this work, the majority voting technique using five classifiers is selected as the best performing method.

The most important findings of this study are the following:

- The best four classified genera are Cantharellus, Coprinus, Pluteus and Russula; however, the classification accuracy of Cantharellus exceeds all other genera significantly due to its very distinct features;
- By downsampling the dataset from ten classes to five classes, the accuracies of each classification model as well as each class are improved significantly;
- The best performing classifiers are the most known and commonly used algorithms, such as random forest, logistic regression, extra tree and linearSVC;
- Classification accuracy is improved by ensemble-based techniques, such as majority voting and mean rule;
- Using the same number of classifiers, the mean rule based method performs better than the majority voting method.

This study provides a basis for mushroom genera determination methods using machine learning. Although classifying mushrooms on the level of genera and species is essential to provide faster and easier determination for mycologists and mushroom hunters, it has been understudied in the past. Therefore, it is crucial to bring attention to the topic and give a starting point for other researchers in the field. By testing different machine learning algorithms and comparing them, this thesis has provided a foundation for researchers to conduct further studies on classifying mushrooms on another level rather than merely based on their edibility and toxicity. This thesis proved that using advanced machine learning algorithms and methods, such as ensemble-based classification, achieves good classification results on smaller datasets. Therefore, this work is an important addition to the studies on the field. In addition to the experimental results obtained, a new mushroom dataset with 1118 labelled images was collected during this study which can be used in the future in similar studies.

Further studies to improve the classification accuracy of mushroom genera need to be carried out. The experiments of the current study were based on grayscale pixels as input to the

algorithms. The next step is to utilise RGB three colour channels as input features to the classifiers in order to improve prediction accuracy. Choosing RGB pixels as features will provide the classification models with more information, such as the specific colour of the cap, stalk and gills, rather than simply the grayscale intensities. Future studies should focus on extracting different features from the mushroom images, such as HOGs and frequency domain. Results of classification models given different feature inputs can be compared to select the best performing ones. Extracting several features also allows combining the result of models with different feature inputs in order to improve the classification workflow. Furthermore, different ensemble methods could be tested on the dataset. In this study, majority voting and mean rule methods were utilised. However, an investigation into other techniques, such as weighted majority voting or median rule, could be carried out. Different ensemble learning methods may provide higher results.

In future research, based on this thesis, it is recommended to carry out background estimation before extracting the features from the images. Background estimation will remove the background making the mushroom more distinguishable, thus, better classifiable. In addition, as the data collection would continue and the sample size grows, machine learning will not be sufficient to handle the amount of data. Therefore, a deep learning framework for mushroom genera classification should be developed. This would also allow to further improve the classification performance as deep learning based architectures usually achieve higher performance due to their higher complexity.

Acknowledgement

I would like to thank my supervisor, Prof. Gholamreza Anbarjafari, for his guidance and support throughout my thesis process. I am also grateful to Eduardo Fabian Garza Garza, without whom I would not have had the dataset to work with and who was ready to talk about mushrooms whenever I asked. I also want to thank Rain Eric Haamer for his help whenever I got stuck on my thesis.

I would like to give my special thanks to my aunt, Reet Karise, who provided me with her critical feedback while writing the thesis. Finally, I am immensely grateful to my family, who supported me mentally and was ready to read my thesis over and over again.

References

[1] P. Cheung, "The nutritional and health benefits of mushrooms", *Nutrition Bulletin*, vol. 35, no. 4, pp. 292–299, 2010.

[2] D. L. Hawksworth and R. Lücking, "Fungal diversity revisited: 2.2 to 3.8 million species", *The fungal kingdom*, pp. 79–95, 2017.

[3] E. Parmasto. (2011). "Eesti seenestik", [Online]. Available: `http://entsyklopeedia.ee/artikkel/eesti_seenestik`. (accessed: 07.03.2021).

[4] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects", *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[5] J. Alzubi, A. Nayyar, and A. Kumar, "Machine learning from theory to algorithms: An overview", in *Journal of Physics: conference series*, IOP Publishing, vol. 1142, 2018, p. 012 012.

[6] P. Maurya and N. P. Singh, "Mushroom classification using feature-based machine learning approach", in *Proceedings of 3rd International Conference on Computer Vision and Image Processing*, Springer, 2020, pp. 197–206.

[7] M. A. Ottom, N. A. Alawad, and K. M. Nahar, "Classification of mushroom fungi using machine learning techniques", *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 5, pp. 2378–2385, 2019.

[8] E. Arnedo Hidalgo *et al.*, "MushroomApp: A mushroom mobile app", 2019.

[9] M. Sulc, L. Picek, J. Matas, T. Jeppesen, and J. Heilmann-Clausen, "Fungi recognition: A practical use case", in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 2316–2324.

[10] K. Kalamees. (2017). "Eesti mükoloogiaühing valis 2017 aasta seene", [Online]. Available: `https://mukoloogiauhing.ut.ee/eesti-m%C3%BCkoloogia%C3%BChing-valis-2017-aasta-seene`. (accessed: 10.05.2021).

[11] K. Kalamees. (Sep. 2008). “Kas minna seenele Kirde-Eestis? Jah, muidugi”, [Online]. Available: http://www.eestiloodus.ee/artikkel2515_2509.html. (accessed: 10.05.2021).

[12] ——, (Sep. 2004). “”Lepaseen” ja leppade tegelikud kaaslased”, [Online]. Available: http://eestiloodus.horisont.ee/artikkel822_818.html. (accessed: 10.05.2021).

[13] J. G. Carbonell, R. S. Michalski, and T. M. Mitchell, “Machine learning: A historical and methodological analysis”, *AI Magazine*, vol. 4, no. 3, pp. 69–79, 1983.

[14] A. M. Turing, “Computing machinery and intelligence”, *MIND: A Quarterly Review of Pyschology and Philosophy*, vol. 59, no. 236, pp. 433–460, 1950.

[15] A. L. Samuel, “Some studies in machine learning using the game of checkers”, *IBM Journal of research and development*, vol. 3, no. 3, pp. 210–229, 1959.

[16] T. M. Mitchell, *Machine Learning*, 1st ed. McGraw-Hill, Inc., 1997.

[17] K. P. Murphy, *Machine learning: A probabilistic perspective*. MIT press, 2012, ch. 1, pp. 1–14.

[18] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*, 2nd ed. MIT press, 2018, ch. 1, pp. 1–7.

[19] S. Wang, W. Chaovalitwongse, and R. Babuska, “Machine learning algorithms in bipedal robot control”, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 5, pp. 728–743, 2012.

[20] K. P. Murphy, *Probabilistic machine learning: An introduction*. MIT Press, 2021. [Online]. Available: probml.ai.

[21] H. Sahli, “An introduction to machine learning”, *TORUS 1–Toward an Open Resource Using Services: Cloud Computing for Environmental Data*, pp. 61–74, 2020.

[22] O. Sagi and L. Rokach, “Ensemble learning: A survey”, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, e1249, 2018.

[23] R. Polikar, “Ensemble learning”, in *Ensemble machine learning*, Springer, 2012, pp. 1–34.

[24] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, “A survey on ensemble learning”, *Frontiers of Computer Science*, vol. 14, no. 2, pp. 241–258, 2020.

[25] J. Goecks, V. Jalili, L. M. Heiser, and J. W. Gray, "How machine learning will transform biomedicine", *Cell*, vol. 181, no. 1, pp. 92–101, 2020.

[26] S. M. McKinney, M. Sieniek, V. Godbole, J. Godwin, N. Antropova, H. Ashrafian, T. Back, M. Chesus, G. S. Corrado, A. Darzi, *et al.*, "International evaluation of an AI system for breast cancer screening", *Nature*, vol. 577, no. 7788, pp. 89–94, 2020.

[27] I. Alrashdi, A. Alqazzaz, E. Aloufi, R. Alharthi, M. Zohdy, and H. Ming, "AD-IoT: Anomaly detection of IoT cyberattacks in smart city using machine learning", in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, 2019, pp. 0305–0310.

[28] P. C. Sen, M. Hajra, and M. Ghosh, "Supervised classification algorithms in machine learning: A survey and review", in *Emerging technology in modelling and graphics*, Springer, 2020, pp. 99–111.

[29] R. G. Morris, T. Martinez, and M. R. Smith, "A hierarchical multi-output nearest neighbor model for multi-output dependence learning", *arXiv preprint arXiv:1410.4777*, 2014.

[30] Y. Feng, M. Zhou, and X. Tong, "Imbalanced classification: An objective-oriented review", *arXiv preprint arXiv:2002.04592*, 2020.

[31] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: A big data - AI integration perspective", *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[32] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Machine learning: A review of classification and combining techniques", *Artificial Intelligence Review*, vol. 26, no. 3, pp. 159–190, 2006.

[33] G. E. Batista and M. C. Monard, "An analysis of four missing data treatment methods for supervised learning", *Applied artificial intelligence*, vol. 17, no. 5-6, pp. 519–533, 2003.

[34] V. Hodge and J. Austin, "A survey of outlier detection methodologies", *Artificial intelligence review*, vol. 22, no. 2, pp. 85–126, 2004.

[35] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "A comparative study of data sampling and cost sensitive learning", in *2008 IEEE International Conference on Data Mining Workshops*, IEEE, 2008, pp. 46–52.

[36] A. Zheng, "Evaluation metrics", in *Evaluating Machine Learning Models*, 2015, pp. 7–13.

[37] R. Entezari-Maleki, A. Rezaei, and B. Minaei-Bidgoli, "Comparison of classification methods based on the type of attributes and sample size.", *J. Convergence Inf. Technol.*, vol. 4, no. 3, pp. 94–102, 2009.

[38] C. Dhaware and K. Wanjale, "Survey on image classification methods in image processing", *International Journal of Computer Science Trends and Technology (IJCST)*, vol. 4, no. 3, pp. 246–248, 2016.

[39] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques", *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, pp. 3–24, 2007.

[40] S. Zhang, Y. Wu, and J. Chang, "Survey of image recognition algorithms", in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, IEEE, vol. 1, 2020, pp. 542–548.

[41] J.-H. Zhao and W.-H. Li, "Improvement intrusion detection based on SVM", in *International Conference on Information Computing and Applications*, Springer, 2012, pp. 53–60.

[42] Y. Zhu, "SVM classification algorithm in ECG classification", in *International Conference on Information Computing and Applications*, Springer, 2012, pp. 797–803.

[43] M. Kulin, T. Kazaz, E. De Poorter, and I. Moerman, "A survey on machine learning-based performance improvement of wireless networks: PHY, MAC and network layer", *Electronics*, vol. 10, no. 3, p. 318, 2021.

[44] L. Breiman, "Random forests", *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[45] J. Sachdeva, V. Kumar, I. Gupta, N. Khandelwal, and C. K. Ahuja, "A package-SFERCB-"Segmentation, feature extraction, reduction and classification analysis by both SVM and ANN for brain tumors"", *Applied soft computing*, vol. 47, pp. 151–167, 2016.

[46] P. Shukla, T. Gupta, A. Saini, P. Singh, and R. Balasubramanian, "A deep learning framework for recognizing developmental disorders", in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2017, pp. 705–714.

[47] M. C. Popescu and L. M. Sasu, "Feature extraction, feature selection and machine learning for image classification: A case study", in *2014 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM)*, IEEE, 2014, pp. 968–973.

[48] S. Alotaibi, S. Sayed, M. Alosaimi, R. Alharthi, A. Banjar, N. Abdulqader, and R. Alhamed, "Pollen molecular biology: Applications in the forensic palynology and future prospects: A review", *Saudi Journal of Biological Sciences*, vol. 27, pp. 1185–1190, 2020.

[49] L. Zhu and P. Spachos, "Towards image classification with machine learning methodologies for smartphones", *Machine Learning and Knowledge Extraction*, vol. 1, no. 4, pp. 1039–1057, 2019.

[50] K. H. Foysal, H. J. Chang, F. Bruess, and J. W. Chong, "Smartfit: Smartphone application for garment fit detection", *Electronics*, vol. 10, no. 1, p. 97, 2021.

[51] Y. Wang, J. Du, H. Zhang, and X. Yang, "Mushroom toxicity recognition based on multi-grained cascade forest", *Scientific Programming*, vol. 2020, 2020.

[52] Ş. Yildirim, "Classification of mushroom data set by ensemble methods", *Recent Innovations in Mechatronics*, vol. 7, no. 1. Pp. 1–4, 2020.

[53] I. S. Al-Mejibli and D. H. Abd, "Mushroom diagnosis assistance system based on machine learning by using mobile devices", *Journal of Al-Qadisiyah for computer science and mathematics*, vol. 9, no. 2, pp. 103–113, 2017.

[54] E. S. Alkronz, K. A. Moghayer, and M. Gazzaz, "Classification of mushroom using artificial neural network", *International Journal of Academic and Applied Research*, vol. 3, pp. 1–5, 2019.

[55] R. Phillips, *Mushrooms*. Macmillan, 2006.

[56] M. E. Ostry, N. A. Anderson, and J. G. O'Brien, *Field guide to common macrofungi in eastern forests and their ecosystem functions*. U.S. Department of Agriculture, 2011.

[57] M. Kuo. (2021). "MushroomExpert", [Online]. Available: `https://www.mushroomexpert.com`. (accessed: 18.03.2021).

[58] R. Polikar, "Ensemble based systems in decision making", *IEEE Circuits and systems magazine*, vol. 6, no. 3, pp. 21–45, 2006.

[59] D. Pilz, *Ecology and management of commercially harvested chanterelle mushrooms*. US Department of Agriculture, Forest Service, Pacific Northwest Research Station, 2003, vol. 576.

[60] D. Drehmel, J.-M. Moncalvo, and R. Vilgalys, "Molecular phylogeny of amanita based on large-subunit ribosomal dna sequences: Implications for taxonomy and character evolution", *Mycologia*, vol. 91, no. 4, pp. 610–618, 1999.

[61] B. Buyck, V. Hofstetter, A. Verbeken, R. Walleyn, *et al.*, "Proposals to conserve or reject names. 1919 proposal to conserve lactarius nom. cons. basidiomycota with a conserved type.", *Tax*, vol. 59, no. 1, pp. 295–296, 2010.

[62] A. Justo, A. M. Minnis, S. Ghignone, N. Menolli, M. Capelari, O. Rodriguez, E. Malysheva, M. Contu, and A. Vizzini, "Species recognition in pluteus and volvopluteus (pluteaceae, agaricales): Morphology, geography and phylogeny", *Mycological Progress*, vol. 10, no. 4, pp. 453–479, 2011.

[63] A. ElRafey and J. Wojtusiak, "Recent advances in scaling-down sampling methods in machine learning", *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 9, no. 6, e1414, 2017.

[64] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, and S. D. Brown, "An introduction to decision tree modeling", *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 18, no. 6, pp. 275–285, 2004.

Non-Exclusive licence to reproduce thesis and make thesis public

I, **Johanna Olesk**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Mushroom genera determination using machine learning

   supervised by Prof., PhD Gholamreza Anbarjafari

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Johanna Olesk
Tartu, 20.05.2021