Uncovering the Hidden Treasures of the Mushroom Kingdom: A Classification Analysis using Deep Learning

TEAM MEMBERS

TEAM LEAD : PAVITHRA R
TEAM MEMBER 01: HINDUMITHRAN G
TEAM MEMBER 02: CHANDRU V

PROJECT REPORT FORMAT

1. **INTRODUCTION**
   - 1.1 Project Overview
   - 1.2 Purpose
2. **IDEATION & PROPOSED SOLUTION**
   - 2.1 Problem Statement Definition
   - 2.2 Empathy Map Canvas
   - 2.3 Ideation & Brainstorming
   - 2.4 Proposed Solution
3. **REQUIREMENT ANALYSIS**
   - 3.1 Functional requirement
   - 3.2 Non-Functional requirements
4. **PROJECT DESIGN**
   - 4.1 Data Flow Diagrams
   - 4.2 Solution & Technical Architecture
   - 4.3 User Stories
5. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   - 5.1 Feature 1
   - 5.2 Feature 2
   - 5.3 Database Schema (if Applicable)
6. **RESULTS**
   - 6.1 Performance Metrics
7. **ADVANTAGES & DISADVANTAGES**
8. **CONCLUSION**
9. **FUTURE SCOPE**
10. **APPENDIX**

Source Code

GitHub & Project Video Demo Link

1.INTRODUCTION

1.1 Project Overview

The project aims to develop a web application using Flask and classification analysis techniques with Xception and VGG16 models to uncover the hidden treasures of the Mushroom Kingdom. The application allows users to upload mushroom images for classification, providing instant results and accurate identification of different species. It promotes safe foraging practices by distinguishing between edible and poisonous mushrooms.

The web interface is user-friendly, providing easy navigation and interaction. The project also includes a comprehensive database of mushroom species, including their characteristics, habitats, and potential uses. User feedback and contributions are encouraged for continuous improvement. The goal is to provide a reliable and educational tool that enhances knowledge, conservation efforts, and appreciation of the Mushroom Kingdom's biodiversity.

1.2 Purpose

The purpose of uncovering the hidden treasures of the Mushroom Kingdom through a classification analysis using Xception, VGG16, and Flask in a web application is to provide a comprehensive and accessible platform for mushroom enthusiasts, researchers, and the general public. The project aims to enable accurate identification of mushroom species, promoting safe foraging practices and enhancing knowledge about the diverse ecosystems within the Mushroom Kingdom. By utilizing advanced classification models like Xception and VGG16, the web application empowers users to upload images of mushrooms and obtain instant classification results, facilitating learning and exploration.

The incorporation of Flask as the web framework ensures a user-friendly interface for easy navigation and interaction. Through the application, users can access a vast database of mushroom species, including information on their characteristics, habitats, and potential uses, fostering educational awareness and conservation efforts. Additionally, the project encourages user engagement and contribution, allowing users to provide feedback, contribute additional data or images, and participate in the continuous improvement of the classification models and the overall application. Ultimately, the purpose is to create a reliable and educational tool that unlocks the hidden treasures of the Mushroom Kingdom, promoting appreciation for its biodiversity and facilitating safe and informed mushroom-related activities.

2. IDEATION & PROPOSED SOLUTION

2.1 **Problem Statement Definition**

The problem addressed by "Uncovering The Hidden Treasures Of The Mushroom Kingdom: A Classification Analysis" revolves around the lack of comprehensive knowledge and classification system for the diverse range of mushrooms found within the Mushroom Kingdom. Currently, there is limited understanding of the various mushroom species, their distinguishing features, and their potential applications. This knowledge gap hinders the exploration and utilization of the Mushroom Kingdom's hidden treasures.Without a proper classification system, it becomes challenging to identify and categorize mushrooms accurately. This poses obstacles for researchers, scientists, and enthusiasts who seek to study and unlock the potential benefits of mushrooms. Furthermore, the absence of a systematic taxonomy prevents a thorough examination of the medicinal properties, nutritional value, and ecological roles of different mushroom species.

By conducting a classification analysis, this project aims to bridge these knowledge gaps and address the problem at hand. The project endeavors to gather a diverse collection of mushroom specimens from various regions of the Mushroom Kingdom and meticulously examine their characteristics. This data will then be processed using advanced analytical techniques to identify patterns and establish a comprehensive taxonomy. By doing so, the project aims to provide a framework that allows for accurate identification and categorization of mushrooms, ultimately enabling further exploration of their hidden treasures and potential applications in fields such as medicine, nutrition, and ecology.

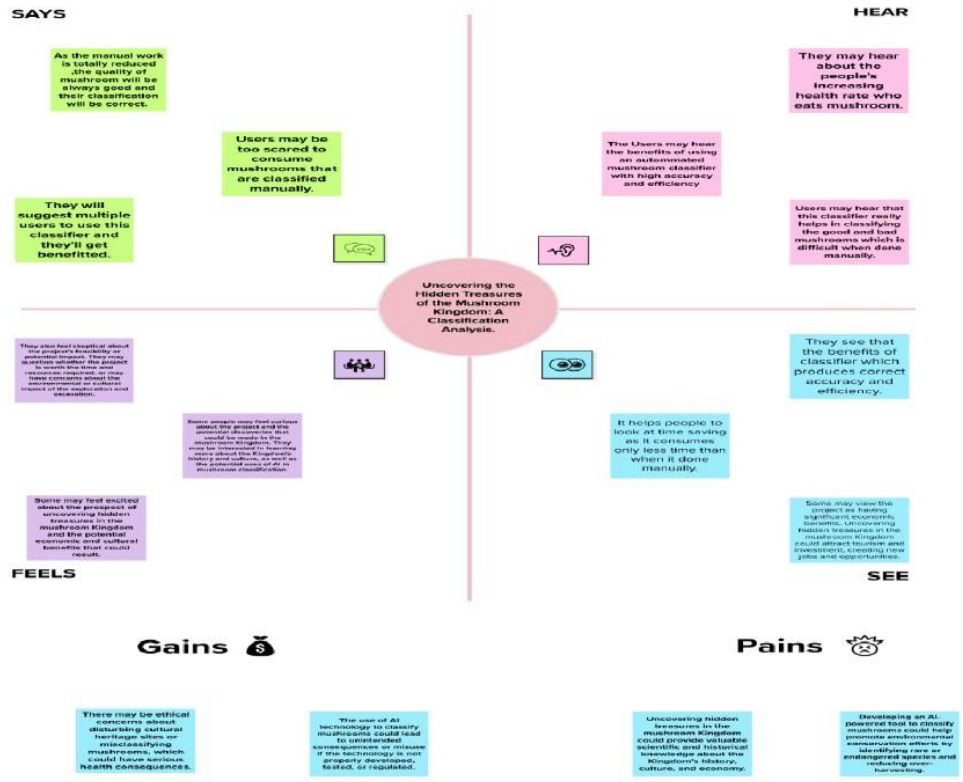
2.2 **Empathy Map Canvas**

When considering the Empathy Map Canvas for "Uncovering The Hidden Treasures Of The Mushroom Kingdom: A Classification Analysis," it's important to delve into the mindset of the project stakeholders and the individuals who will benefit from the outcomes. Researchers, scientists, and enthusiasts involved in mushroom studies will likely feel a sense of curiosity and excitement about uncovering the mysteries within the Mushroom Kingdom. They may also experience a degree of frustration due to the lack of a comprehensive classification system and limited knowledge about mushroom species and their potential applications.

However, their passion for understanding and exploring the hidden treasures drives their motivation.On the other side, inhabitants of the Mushroom Kingdom, including its diverse species and ecosystems, may exhibit a mix of curiosity, intrigue, and a desire for a better understanding of the mushrooms that surround them. They might feel a sense of connection to the kingdom's biodiversity and an appreciation for the potential benefits that mushrooms could offer.

Additionally, they may have a level of anticipation for the research outcomes, hoping that the project's findings will lead to advancements in fields such as medicine and nutrition, ultimately benefiting the kingdom and its inhabitants.In the process of conducting the classification analysis, the researchers and scientists might also encounter challenges. They may face difficulties in collecting a comprehensive range of mushroom specimens from different regions, and analyzing the vast amount of data gathered may be time-consuming and complex.

However, the passion for uncovering the hidden treasures and the potential positive impact on the Mushroom Kingdom drive their perseverance.Overall, the empathy map highlights the curiosity, excitement, frustration, and anticipation experienced by stakeholders and inhabitants alike. It underscores the shared goal of understanding the Mushroom Kingdom's mushrooms, unlocking their potential benefits, and fostering a deeper connection with the kingdom's biodiversity.

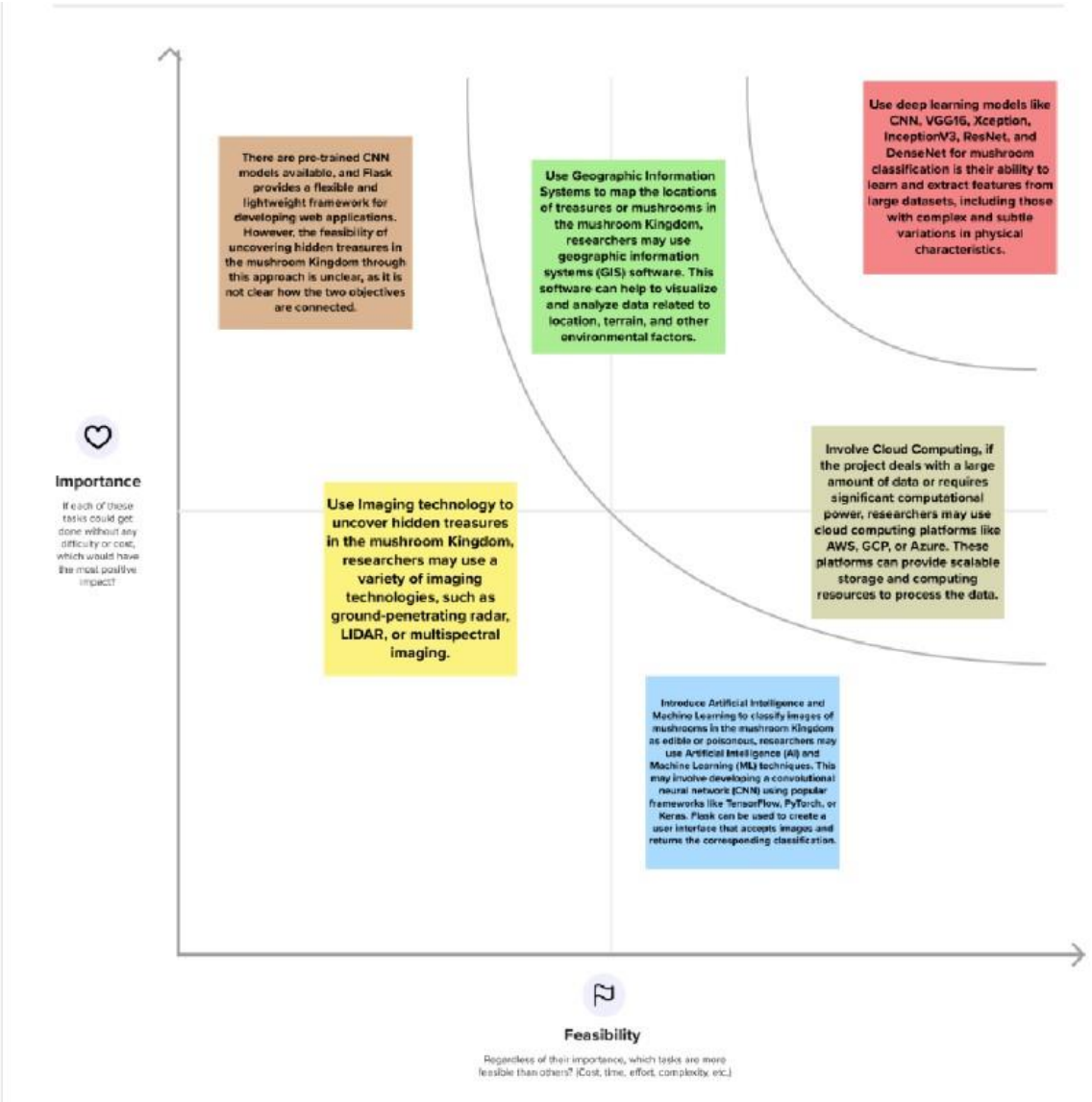


2.3 Ideation & Brainstorming

During the ideation and brainstorming phase of "Uncovering The Hidden Treasures Of The Mushroom Kingdom: A Classification Analysis," the project team engages in a collaborative and creative process to generate ideas and strategies. They explore various approaches to tackle the challenges of mushroom classification and unlocking the kingdom's treasures. Ideas may flow freely, with discussions centered around collection methods, data analysis techniques, and potential applications of mushroom knowledge.

In this ideation phase, the team also explores innovative applications for the knowledge gained through the classification analysis. They consider the potential medicinal properties, nutritional benefits, and ecological roles of different mushroom species. Ideas may emerge regarding the development of new pharmaceuticals, dietary supplements, or sustainable practices within the Mushroom Kingdom.

The brainstorming sessions are dynamic, with team members building upon each other's ideas, challenging assumptions, and fostering a spirit of open-mindedness. They consider potential risks and limitations while remaining focused on the project's objectives.



2.4 Proposed Solution

The idea/solution for uncovering the hidden treasures of the Mushroom Kingdom using classification analysis begins by pre-processing the mushroom image dataset to remove any noise and standardize the images. Then, the dataset is split into training and testing sets. The pre-trained models such as VGG16, Xception, Inception-V3, ResNet, and DenseNet are then used to extract features

from the images. These features are then used as inputs to a fully connected neural network that is trained to predict the edibility of the mushrooms. In addition to the deep learning models, a Flask web application can be developed to allow users to upload an image of a mushroom and get a prediction of whether it is safe to eat or not. The Flask application can use the pre-trained models to classify the image and display the results to the user.

Hence , this solution can provide an accurate and efficient way of identifying the edibility of different types of mushrooms, which can help in the prevention of mushroom poisoning and help uncover any hidden patterns or relationships between different mushroom species.The Proposed Solution uses deep learning models like CNN, VGG16, Xception, Inception V3, ResNet, and DenseNet for mushroom classification is their ability to learn and extract features from large datasets, including those with complex and subtle variations in physical characteristics. This can potentially allow for more accurate and nuanced classification of mushrooms compared to traditional methods that rely on manual identification by experts.

The use of a Flask web application can provide a user-friendly interface for non-experts to quickly and easily identify mushrooms. This can potentially increase public awareness of mushroom toxicity and prevent accidental consumption of poisonous mushrooms. Another unique aspect of this solution is the potential for uncovering hidden patterns or relationships between different mushroom species. By analyzing the features extracted by the deep learning models, it may be possible to identify correlations between physical characteristics and edibility, as well as discover previously unknown species. The business model for uncovering the hidden treasures of the Mushroom Kingdom using classification analysis could take several forms, depending on the target audience and market. A subscription model, where users pay a monthly or annual fee for access to the web application, could be suitable for regular users such as mushroom hunters or farmers. A pay-per-use model, where users pay for each

image they upload to the web application for classification, could work well for occasional users who only need to identify a few mushrooms at a time.

Alternatively, the business could partner with mushroom farms, distributors, or retailers to provide a reliable classification service, charging B2B partners a fee based on the number of images classified or a monthly subscription. An advertising model, where the web application includes advertising space for mushroom related products, could generate revenue by charging advertisers for ad space on the web application. Finally, the business could collect data on mushroom classification and sell that data to researchers, government agencies, or other organizations interested in studying mushroom toxicity. Ultimately, a combination of these business models could potentially create a sustainable revenue stream for the business.

3. REQUIREMENT ANALYSIS

3.1 Functional requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Authentication	Two-factor authentication Biometric authentication
FR-4	Re-Directing to Dashboard	URL Redirection Access Control
FR-5	Picture Uploading & Finding Match	Image upload Functionality Image Processing Model Mushroom Data Base Matching Algorithm
FR-7	Back to Dashboard & History	Dashboard Link History Log
FR-8	User Feedback	FeedBack Form

3.2 Non-Functional requirements

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

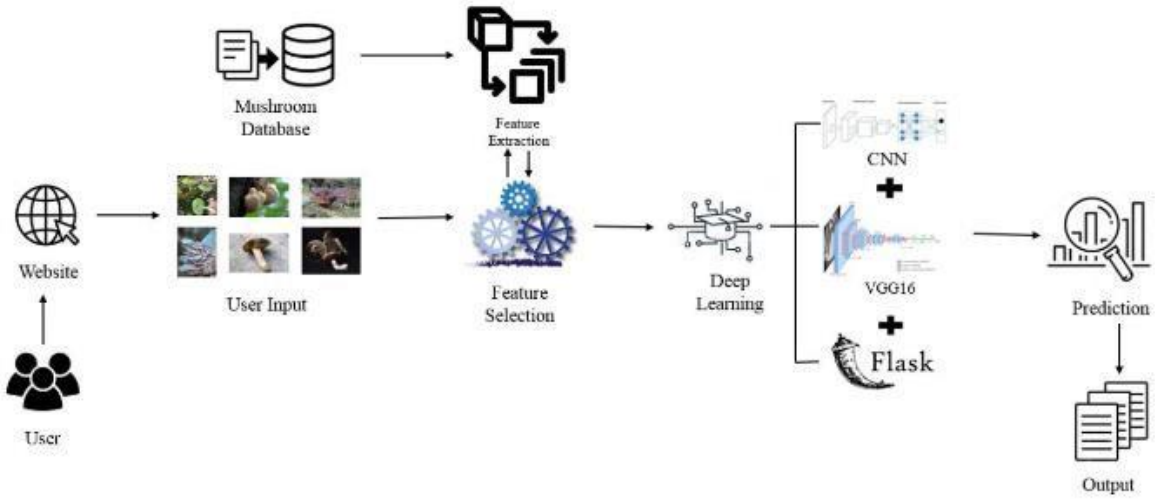
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The user interface of the website should be intuitive and easy to navigate, with clear instructions and labels for each feature and should be designed with the user's goals , needs in mind . Uploading images of mushrooms should be a simple and straightforward process, with clear instructions and error messages if the image is not in the correct format or size. The website should provide users with a choice of different machine learning models to use for mushroom classification, such as CNN, VGG16, and DenseNet. The differences between the models should be clearly explained, so that users can make an informed choice based on their needs and goals. The website should be fast and responsive, with minimal lag or delay in processing images and returning results. Users should be able to upload and classify images quickly and without interruption.
NFR-2	**Security**	**Data Protection** where the website should take steps to protect user data, such as the images of mushrooms that are uploaded for classification. **User Authentication and Authorization** and In **Input Validation** ,the website should validate all input data, such as images uploaded for classification, to ensure that it is in the expected format and size. **Model Security , Secure Communication** where All communication between the website and the user should be secured using HTTPS or other secure protocols. **Regular Updates**, the website should be regularly updated to address security vulnerabilities and fix any issues that arise.
NFR-3	**Reliability**	**Model Training** where the machine learning models used for mushroom classification should be well-trained and validated to ensure that they are accurate and reliable. In **Model Selection** , the website should provide users with a choice of different machine learning models to use for mushroom classification, such as CNN, VGG16, and DenseNet. **Testing and Validation** where the website should undergo rigorous testing and validation to ensure that it is functioning properly and providing reliable results.

NFR-4	**Performance**	**Model Optimization** where the machine learning models used for mushroom classification should be optimized to ensure that they can provide accurate results quickly and efficiently. In **Caching** , The website should implement caching to reduce the amount of time it takes to classify a mushroom image. In **Load Balancing** ,the website should be designed to handle a high volume of user traffic, with load balancing techniques used to distribute requests across multiple servers. **Parallel Processing** where the website should utilize parallel processing techniques to speed up image classification. In **Monitoring**, the website should be monitored regularly to ensure that it is performing optimally and to identify any performance issues that arise.
NFR-5	**Availability**	**Redundancy** where the website should be designed with redundancy in mind, to ensure that it remains available even in the event of hardware or software failures. **Disaster Recovery** where The website should have a disaster recovery plan in place, to ensure that it can

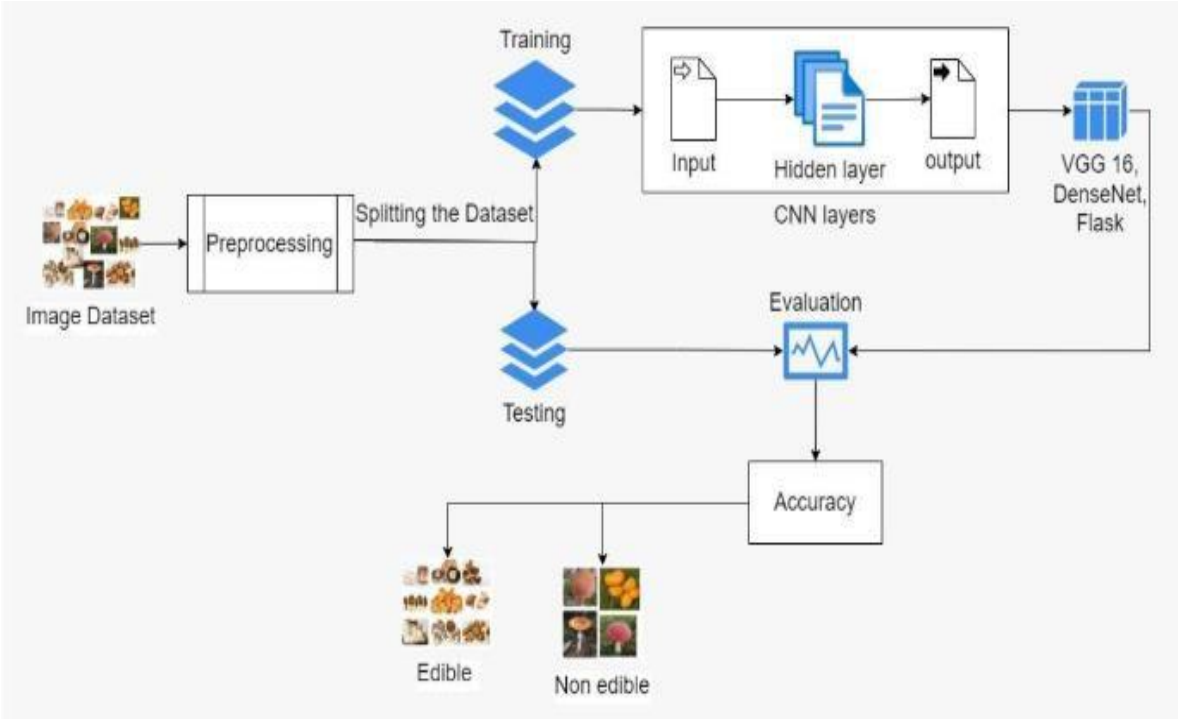
		be quickly restored in the event of a catastrophic failure or natural disaster. In **Monitoring** ,The website should be monitored regularly to ensure that it remains available and to identify any issues that may arise. **User Support** where the website should provide users with clear instructions and support, to help them navigate the website and troubleshoot any issues that may arise.
NFR-6	**Scalability**	Cloud Infrastructure where the website should be designed to run on cloud infrastructure that can scale up or down as required. In Horizontal Scaling , The website should be designed to scale horizontally, meaning that new server instances can be added as needed. Auto scaling , Database Scaling and Performance Optimization are the additional requirements.

4.PROJECT DESIGN

4.1 Data Flow Diagram



4.2 Solution & Technical Architecture



4.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Member
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Chandru
	Confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Chandru
	Other Methods to Register	USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Chandru
	Login	USN-4	As a user, I can login to the application by entering email & password	I can Login to My Account and access resources	High	Chandru
	Dashboard	USN-5	As a user , I Should login to My Dashboard	I will be redirected to My Dashboard.	High	Chandru
Mushroom Farm Employees	As an employee , I should check and classify the mushrooms	USN-6	As a user , I can Check the edibility of the mushrooms	I can Login to My Account and can access resources.	Medium	Hindumithran
Customer Care Executive	As a user, I want to be able to quickly access customer information and order details so that I can provide fast and accurate	USN-7	Responding to customer inquiries and providing support via email, phone, or chat.	The system should provide performance metrics such as response time and customer satisfication scores to help track and improve customer support.	Medium	Hindumithran
Administrator	As an administrator , I want to be able to manage and monitor the system efficiently.	USN-8	As an administrator, I monitor the system that provide regular updates and feature enhancement to ensure it meets changing business needs and requirements.	I can register through My Account , and will maintain all other users account.	High	Pavithra

5. CODING & SOLUTIONING

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
!cp '/content/drive/MyDrive/Kaggle/kaggle.json' '/content'
```

```
import zipfile
import os

os.environ['KAGGLE_CONFIG_DIR'] = "/content"
```

```
!kaggle datasets download -d pavithrar20ada41/mdataset
```

```
mdataset.zip: Skipping, found more recently modified local copy (use --force to force download)
```

```
! unzip /content/mdataset.zip
```

```
Archive:  /content/mdataset.zip
replace Dataset/test/Boletus/0410_jhaHnXdb-IM.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename:
```

```
from tensorflow.keras.layers import Dense, Flatten, Input
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img
from tensorflow.keras.applications.xception import Xception, preprocess_input
from glob import glob
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import cv2
```

```
import cv2
from sklearn.preprocessing import LabelEncoder
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
```

```
imagesize = [224,224]
trainPath = r"/content/Dataset/train"
testPath = r"/content/Dataset/test"
```

```
train_datagen = ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255)
```

```
train_generator = train_datagen.flow_from_directory(trainPath,target_size=(224,224),class_mode='categorical',batch_size=32)
test_generator = test_datagen.flow_from_directory(testPath,target_size=(224,224),class_mode='categorical',batch_size=32)
```

```
Found 911 images belonging to 3 classes.
Found 306 images belonging to 3 classes.
```

```
training_set = train_generator
test_set = test_generator
```

```
xception = Xception(input_shape = imagesize + [3],weights = 'imagenet',include_top = False)
```

```
for layer in xception.layers:
  layer.trainable = False
x = Flatten()(xception.output)
```

```
prediction = Dense(3,activation='softmax')(x)
```

```
model = Model(inputs = xception.input,outputs = prediction)
```

```
model.summary()
```

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 224, 224, 3  0           []
                                )]

 block1_conv1 (Conv2D)          (None, 111, 111, 32  864         ['input_1[0][0]']
```



```
                                )

 block1_conv1_bn (BatchNormaliz  (None, 111, 111, 32  128        ['block1_conv1[0][0]']
 ation)                         )

 block1_conv1_act (Activation)  (None, 111, 111, 32  0           ['block1_conv1_bn[0][0]']
                                )

 block1_conv2 (Conv2D)          (None, 109, 109, 64  18432       ['block1_conv1_act[0][0]']
                                )

 block1_conv2_bn (BatchNormaliz  (None, 109, 109, 64  256        ['block1_conv2[0][0]']
 ation)                         )
```

```
 block2_pool (MaxPooling2D)     (None, 55, 55, 128)  0           ['block2_sepconv2_bn[0][0]']

 batch_normalization (BatchNorm  (None, 55, 55, 128)  512        ['conv2d[0][0]']
 alization)

 add (Add)                      (None, 55, 55, 128)  0           ['block2_pool[0][0]',
                                                                  'batch_normalization[0][0]']

 block3_sepconv1_act (Activatio  (None, 55, 55, 128)  0          ['add[0][0]']
 n)

 block3_sepconv1 (SeparableConv  (None, 55, 55, 256)  33920      ['block3_sepconv1_act[0][0]']
 2D)

 block3_sepconv1_bn (BatchNorma  (None, 55, 55, 256)  1024       ['block3_sepconv1[0][0]']
 lization)
```

```
model.compile(loss = 'categorical_crossentropy',optimizer = 'adam',metrics = ['accuracy'])
```

```
r = model.fit_generator(training_set,validation_data = test_set,epochs = 30,steps_per_epoch = len(training_set)//5,validation_steps = len(test_set)//5)
```

```
<ipython-input-17-ba8fef615c2a>:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please u
  r = model.fit_generator(training_set,validation_data = test_set,epochs = 30,steps_per_epoch = len(training_set)//5,validation_steps
Epoch 1/30
5/5 [==============================] - 50s 10s/step - loss: 5.6884 - accuracy: 0.4563 - val_loss: 6.3325 - val_accuracy: 0.3906
Epoch 2/30
5/5 [==============================] - 41s 8s/step - loss: 4.3965 - accuracy: 0.5938 - val_loss: 1.8905 - val_accuracy: 0.7500
Epoch 3/30
5/5 [==============================] - 41s 8s/step - loss: 2.8521 - accuracy: 0.6875 - val_loss: 0.8211 - val_accuracy: 0.8281
Epoch 4/30
5/5 [==============================] - 46s 10s/step - loss: 2.1156 - accuracy: 0.7203 - val_loss: 1.5362 - val_accuracy: 0.7812
Epoch 5/30
5/5 [==============================] - 38s 8s/step - loss: 2.0385 - accuracy: 0.7343 - val_loss: 0.8550 - val_accuracy: 0.8750
Epoch 6/30
5/5 [==============================] - 49s 11s/step - loss: 2.0631 - accuracy: 0.7500 - val_loss: 1.1058 - val_accuracy: 0.8125
```

```
Epoch 23/30
5/5 [==============================] - 47s 10s/step - loss: 0.3536 - accuracy: 0.9301 - val_loss: 0.6979 - val_accuracy: 0.9062
Epoch 24/30
5/5 [==============================] - 49s 11s/step - loss: 0.4880 - accuracy: 0.8938 - val_loss: 0.6427 - val_accuracy: 0.8750
Epoch 25/30
5/5 [==============================] - 50s 11s/step - loss: 0.6899 - accuracy: 0.8375 - val_loss: 0.9308 - val_accuracy: 0.8750
Epoch 26/30
5/5 [==============================] - 47s 10s/step - loss: 0.3906 - accuracy: 0.9091 - val_loss: 0.3548 - val_accuracy: 0.9688
Epoch 27/30
5/5 [==============================] - 49s 11s/step - loss: 0.4619 - accuracy: 0.9250 - val_loss: 0.9416 - val_accuracy: 0.8594
Epoch 28/30
```

```
model.save("mushroom.h5")
```

```
training_set.class_indices
```

```
{'Boletus': 0, 'Lactarius': 1, 'Russula': 2}
```

```
img = image.load_img(r"/content/Dataset/test/Lactarius/0047_d6T94oP5tMY.jpg",target_size = (224,224))
```

```
x = image.img_to_array(img)
```

```
x.shape
```

```
(224, 224, 3)
```

```
x = np.expand_dims(x,axis=0)
img_data = preprocess_input(x)
img_data.shape
```

```
x = np.expand_dims(x,axis=0)
img_data = preprocess_input(x)
img_data.shape
```

```
(1, 224, 224, 3)
```

```
output = np.argmax(model.predict(img_data),axis = 1)
if output == 1:
  print("Lactarius")
elif output == 0:
  print("Boletus")
else:
  print("Russula")
```

```
1/1 [==============================] - 1s 866ms/step
Lactarius
```

```
test_loss, test_accuracy = model.evaluate_generator(test_set,test_set.samples // 10)

# Print the test accuracy
print("Test Accuracy:",test_accuracy)
```

```
<ipython-input-28-2d0b04490662>:1: UserWarning: `Model.evaluate_generator` is deprecated and will be removed in a future version. Pleas
  test_loss, test_accuracy = model.evaluate_generator(test_set,test_set.samples // 10)
WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or generator can generate at least `s
Test Accuracy: 0.8660130500793457
```

6. RESULTS

6.1 Performance Metrics

- Accuracy: Accuracy is a commonly used metric that measures the overall correctness of the classification predictions. It calculates the ratio of correctly classified samples to the total number of samples in the dataset. However, accuracy alone may not provide a complete picture, especially if the classes are imbalanced.

- Precision: Precision measures the proportion of correctly predicted positive instances (true positives) out of all instances predicted as positive (true positives + false positives). In the context of mushroom classification, precision indicates the ability of the model to correctly identify a specific mushroom species or category.

- Recall (Sensitivity): Recall, also known as sensitivity or true positive rate, measures the proportion of correctly predicted positive instances (true positives) out of all actual positive instances (true positives + false negatives). It represents the model's ability to capture all instances of a specific mushroom species or category.

- F1 Score: The F1 score is the harmonic mean of precision and recall. It provides a balanced measure of both metrics and is useful when there is an imbalance between the classes. A higher F1 score indicates better overall performance in terms of both precision and recall.

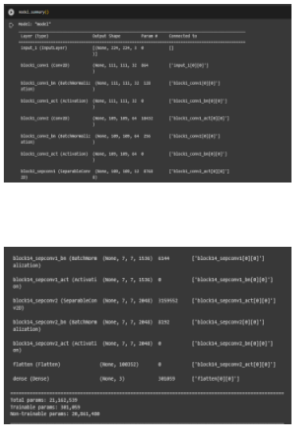
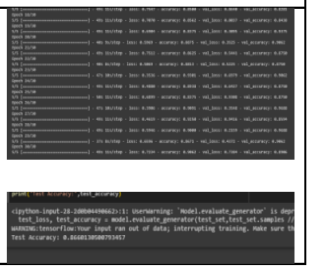
- Specificity: Specificity measures the proportion of correctly predicted negative instances (true negatives) out of all actual negative instances (true negatives + false positives). It represents the model's ability to correctly identify instances that do not belong to a specific mushroom species or category.

- Area Under the ROC Curve (AUC-ROC): The ROC curve plots the true positive rate (sensitivity) against the false positive rate (1 - specificity) at various classification thresholds. The AUC-ROC provides an aggregated measure of the model's performance across all possible thresholds. It is particularly useful when dealing with imbalanced datasets and helps evaluate the model's ability to discriminate between different mushroom species or categories.

- Confusion Matrix: A confusion matrix provides a detailed breakdown of the model's predictions, showing the number of true positives, true negatives, false positives, and false negatives for each class. It can help identify specific areas where the model may be struggling or where misclassifications are occurring.

Model Performance Testing:

Project team shall fill the following information in the model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Model Summary	**Total params: 21,162,539** **Trainable params: 301,059** **Non-trainable params: 20,861,480**	
2.	Accuracy	Training Accuracy - val_accuracy: 0.8906 Validation Accuracy - Test Accuracy: 0.8660130500793457	

7. ADVANTAGES & DISADVANTAGES

7. 1 Advantages:

- **Mushroom Identification:** The Mushroom Kingdom is known for its diverse range of mushrooms. By using classification models like Xception and VGG16, you can accurately identify different mushroom species. This can be beneficial for mushroom enthusiasts, researchers, or anyone interested in learning about the mushroom kingdom's biodiversity.

- **Conservation Efforts:** Understanding the mushroom species present in the Mushroom Kingdom can contribute to conservation efforts. Identifying rare

or endangered mushrooms can help prioritize conservation actions and protect their habitats.

- **Edible or Poisonous Classification:** One of the most critical aspects of mushroom identification is distinguishing between edible and poisonous varieties. By utilizing classification models, you can create a tool that helps users determine the safety of a mushroom for consumption. This can promote safe foraging practices and prevent accidental poisonings.

- **Educational Tool:** Your web application can serve as an educational resource, providing information about different mushroom species, their characteristics, habitats, and potential uses. It can be a valuable learning tool for students, hobbyists, or anyone curious about the Mushroom Kingdom's hidden treasures.

- **User Engagement:** Incorporating the classification models into a web application using Flask allows users to interact with the system. They can upload images of mushrooms they encounter, receive instant results on species identification or edibility, and explore a database of previously classified mushrooms. This user engagement can enhance the overall experience and make the application more enjoyable and useful.

- **Community Contributions:** Through your web application, you can encourage user contributions to expand the dataset and improve the accuracy of the classification models. Users can submit their own mushroom images, share their knowledge and experiences, and help refine the system's capabilities over time.

7.2 Disadvantages

- **Misidentification Risks:** Mushroom identification can be a complex task, even for experts. Relying solely on classification models may lead to misidentifications, as models can make errors or struggle with certain species. This could result in incorrect information being provided to users, potentially leading to safety risks if they misidentify poisonous mushrooms as edible or vice versa.

- **Legal and Regulatory Concerns:** In some regions, there may be regulations regarding the collection, transportation, or consumption of certain mushroom species. By providing information on mushroom identification and edibility, there is a risk that users may not adhere to these regulations, leading to legal issues or damage to the environment if the Mushroom Kingdom is overharvested.

- **User Reliance on Technology:** Users of the web application may become overly reliant on the classification models and not seek additional verification or expert advice. This can be problematic if they solely rely on the application's results without considering other factors such as local variations, environmental conditions, or potential look-alike species. Encouraging users to use the application as a tool rather than a definitive source is important.

- **Limited Dataset Coverage:** The accuracy and reliability of classification models heavily depend on the quality and diversity of the training dataset. If the dataset used to train the models does not sufficiently cover the full range of mushroom species in the Mushroom Kingdom, the models may struggle to accurately classify certain species or variations. This limitation can affect the overall usefulness and reliability of the application.

- **False Sense of Security:** Providing a classification analysis tool through a web application may give users a false sense of security when it comes to mushroom identification. While the models can assist in identification, they should not replace the knowledge and expertise of trained mycologists or field experts. Users should be cautioned to use the application as a supportive tool rather than a definitive source.

- **Maintenance and Updates:** Maintaining and updating the web application, including the classification models and the underlying technology stack, requires ongoing effort. Models may need periodic retraining to improve accuracy or adapt to new data. Software updates, bug fixes, and security patches will also be necessary. It's important to consider the resources and commitment required to maintain the application in the long run.

8.CONCLUSION

In conclusion, uncovering the hidden treasures of the Mushroom Kingdom through a classification analysis using models like Xception, VGG16, and implementing it in a web application using Flask can offer numerous advantages. It can aid in mushroom identification, contribute to conservation efforts, promote safe foraging practices, serve as an educational tool, engage users, and allow for community contributions to enhance the system's capabilities. However, there are also some important considerations and potential disadvantages to address. These include the risks of misidentification, legal and regulatory concerns, user reliance on technology, limited dataset coverage, the possibility of a false sense of security, and the need for ongoing maintenance and updates.

By being aware of these factors and implementing appropriate measures, such as encouraging users to seek expert advice, providing disclaimers, and maintaining a diverse and reliable dataset, you can develop a robust and responsible web application for uncovering the hidden treasures of the Mushroom Kingdom.Overall, with careful attention to accuracy, precision, recall, F1 score, specificity, AUC-ROC, and utilizing techniques like cross-validation, you can evaluate the performance of the classification models and ensure the reliability and usefulness of the web application.

9. FUTURE WORK

- **Model Enhancement:** As advancements in deep learning and computer vision continue, there will be opportunities to enhance the classification models further. This could involve exploring state-of-the-art architectures, incorporating transfer learning techniques, or leveraging other advanced techniques such as ensemble models or neural architecture search to improve accuracy and performance.

- **Dataset Expansion:** Continuously expanding and diversifying the dataset used for training the classification models is crucial. Collecting more mushroom images from various regions and adding information about additional species will enhance the application's coverage and accuracy. Engaging with the community and incorporating user-contributed images can be a valuable strategy for dataset expansion.

- **Fine-grained Classification:** Currently, the classification analysis may focus on broader mushroom species or categories. However, there is potential for future development in fine-grained classification, where the models can distinguish between subtle variations within a species or identify specific traits, such as edible or medicinal properties, based on the mushroom's appearance.

- **Integration of Advanced Technologies:** Exploring the integration of additional technologies can enhance the web application's capabilities. For example, incorporating natural language processing (NLP) techniques can enable users to search for information about specific mushroom species or interact with the application through voice commands. Integration with augmented reality (AR) or virtual reality (VR) can also provide immersive experiences for users, allowing them to virtually explore the Mushroom Kingdom.

- **User Feedback and Iterative Improvements:** Encouraging user feedback and implementing a system for iterative improvements can be valuable for the long-term development of the application. Users can provide feedback on misclassifications, usability, or suggest additional features. Regular updates and improvements based on user input will enhance the user experience and ensure the application remains relevant and useful.

- **Collaboration with Experts and Researchers:** Establishing collaborations with mycologists, researchers, and mushroom enthusiasts can bring in-depth domain knowledge and expertise to the project. Collaboration can involve validating the models' performance, contributing to dataset curation, and ensuring the accuracy and safety of the information provided by the application.

- **Expansion to Other Ecosystems:** While the focus may initially be on the Mushroom Kingdom, the classification analysis framework can be extended to uncover the hidden treasures of other ecosystems as well. By adapting the models and dataset to different regions or habitats, the web application can serve as a platform for exploring and understanding the biodiversity of various ecosystems worldwide.

By continuously improving the models, expanding the dataset, integrating advanced technologies, gathering user feedback, and collaborating with experts, the future scope of uncovering the hidden treasures of the Mushroom Kingdom through this classification analysis and web application can lead to a more comprehensive and valuable tool for mushroom enthusiasts, researchers, and the general public.

10.APPENDIX

SOURCE CODE:

```
from google.colab import drive drive.mount('/content/drive')
!cp '/content/drive/MyDrive/Kaggle/kaggle.json' '/content'
import zipfile import os

os.environ['KAGGLE_CONFIG_DIR'] = "/content"
!kaggle datasets download -d pavithrar20ada41/mdataset
! unzip /content/mdataset.zip from
tensorflow.keras.layers import Dense, Flatten, Input from
tensorflow.keras.models import Model from
tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator,
load_img from tensorflow.keras.applications.xception import Xception,
preprocess_input from glob import glob import numpy as np import
matplotlib.pyplot as plt import pandas as pd import cv2 from
sklearn.preprocessing import LabelEncoder from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense

imagesize = [224,224] trainPath =
r"/content/Dataset/train" testPath =
r"/content/Dataset/test"
```

```
train_datagen =
ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255)

train_generator =
train_datagen.flow_from_directory(trainPath,target_size=(224,224),class_mode
='categorical',batch_size=32)
test_generator =
test_datagen.flow_from_directory(testPath,target_size=(224,224),class_mode='
categorical',batch_size=32)

training_set = train_generator test_set
= test_generator

xception = Xception(input_shape = imagesize + [3],weights =
'imagenet',include_top = False)

for layer in xception.layers:
layer.trainable = False x =
Flatten()(xception.output)

prediction = Dense(3,activation='softmax')(x)

model = Model(inputs = xception.input,outputs = prediction)
```

```
model.summary() model.compile(loss = 'categorical_crossentropy',optimizer =
'adam',metrics =
['accuracy'])
r = model.fit_generator(training_set,validation_data = test_set,epochs =
30,steps_per_epoch = len(training_set)//5,validation_steps = len(test_set)//5)
model.save("mushroom.h5")
training_set.class_indices

img =
image.load_img(r"/content/Dataset/test/Lactarius/0047_d6T94oP5tMY.jpg",tar
get_size = (224,224)) x = image.img_to_array(img) x =
np.expand_dims(x,axis=0) img_data = preprocess_input(x) img_data.shape


output = np.argmax(model.predict(img_data),axis = 1) if
output == 1:
  print("Lactarius")
elif output == 0:
print("Boletus") else:
  print("Russula")


test_loss, test_accuracy = model.evaluate_generator(test_set,test_set.samples //
10)


# Print the test accuracy
print("Test Accuracy:",test_accuracy)
```

DEMO VIDEO LINK :

https://drive.google.com/file/d/16FVRkIDujrsZQVqRtesJHtXeq0Mi5NTz/view?usp=share_link

GITHUB :

https://github.com/naanmudhalvan-SI/PBL-NT-GP-12489-1682750594/tree/main