High Level Design (HLD)

Mushroom Type Classifier

Revision No: 1.0

Last Date of Revision: 25/05/2023.

Document Version Control-

Date Issued	Version	Description	Author
25/05/2023	1.0	HLD – V- 1.0	Mahesh. A

Contents -

Description	Page No.
Document Version Control	2
1. Introduction	4
1.1 Necessity of HLD	4
2. General Description	4
2.1 Product Perspective	4
2.2 Problem Statement	4
2.3 Data Requirements	4
2.4 The format would as the below mentioned sample	6
2.5 Tools Required	7
3. Design Details	8
3.1 Process Flow-	8
3.2 Model Training and Evaluation-	8
3.3 Deployment Process	9
3.4 Event Log	9
3.5 Error Handling	9
4 Performance-	
4.1 Reusability	10
4.2 Application Compatibility	10
4.3 Resource Utilization	10
4.4 Deployment	10
5 Conclusion	

1 Introduction

1.1Necessity of HLD–

The purpose of this document is to add the necessary information to the current project for Modelling and its coding. This can be treated as a user manual.

2. General Description

2.1 Product Perspective-

This mushroom classifier is the project is based on the Machine learning Classifier which helps us to identify the Mushrooms class.

2.2 Problem S-

To create the AL solution for the mushroom classification

. To detect to Mushroom as Edible or poisonous

2.3 Data Requirements-

The data requirements completely depend on the problem statement.

We need the data in the .CSV or any other pandas readable format with the below mentioned details.

Attribute Information: (classes: edible=e, poisonous=p)

a. cap-shape: bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s

b. cap-surface: fibrous=f, grooves=g, scaly=y, smooth=s

c. cap-color: brown=n, buff=b, cinnamon=c, gray =g, green=r, pink=p, purple=u, red=e, white=w, yellow=y

d. bruises: bruises=t, no=f

e. odor: almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m, none=n, pungent=p, spicy=s

f. gill-attachment: attached=a, descending=d, free=f, notched=n

g. gill-spacing: close=c, crowded=w, distant=d

h. gill-size: broad=b, narrow=n

i. gill-color: black=k, brown=n, buff=b, chocolate=h, gray=g, green=r, orange=o, pink=p, purple=u, red=e, white=w, yellow=y

j. stalk-shape: enlarging=e, tapering=t

k. Stalk-root: bulbous=b, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r, missing=?

l. stalk-surface-above-ring: fibrous=f, scaly=y, silky=k, smooth=s

m. stalk-surface-below-ring: fibrous=f, scaly=y, silky=k, smooth=s

n. stalk-color-above-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y

o. stalk-color-below-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y

p. veil-type: partial=p, universal=u

q. veil-color: brown=n, orange=o, white=w, yellow=y

r. ring-number: none=n, one=o, two=t

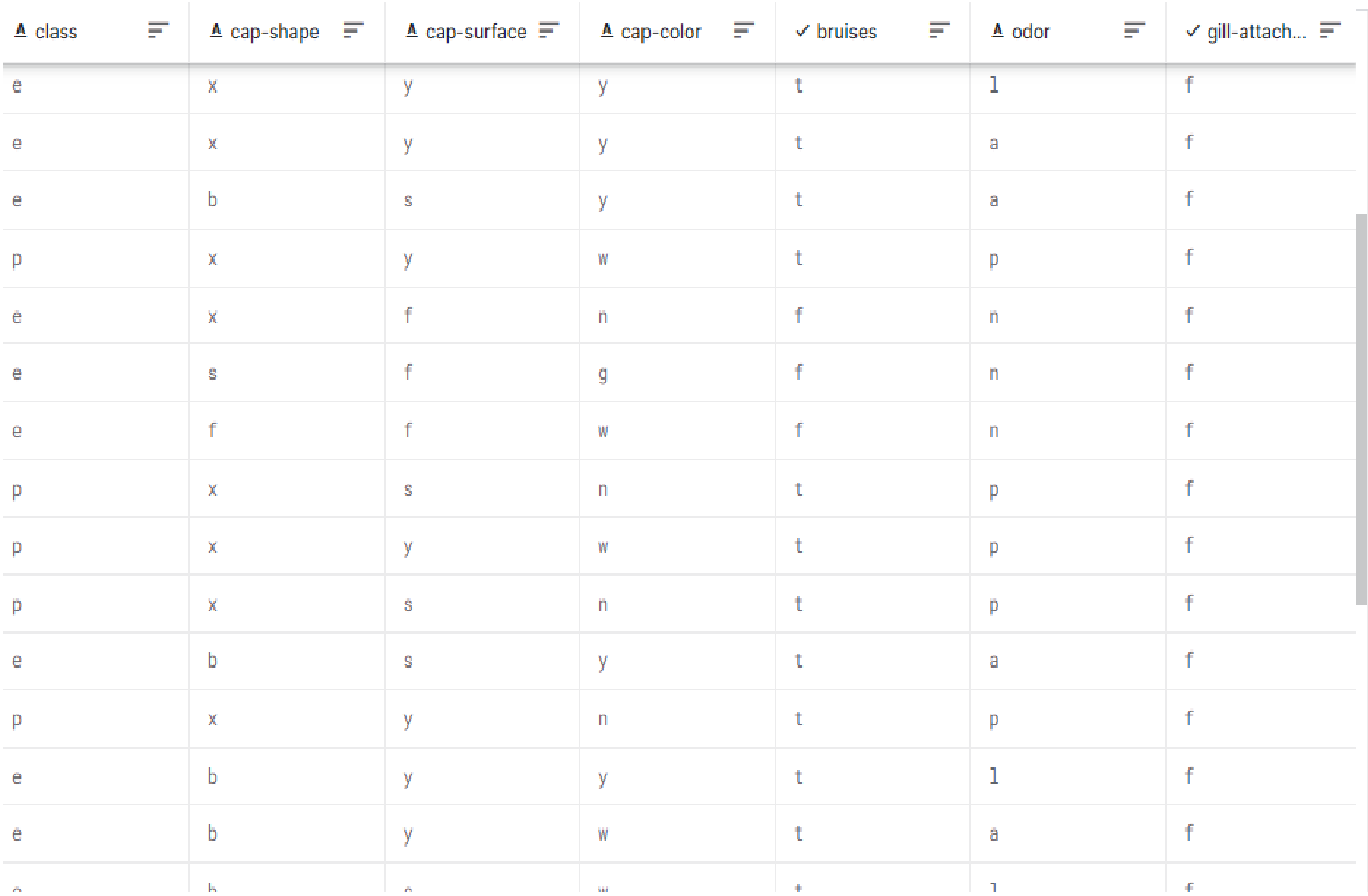
s. ring-type: cobwebby=c, evanescent=e, flaring=f, large=l, none=n, pendant=p, sheathing=s, zone=z

t. spore-print-color: black=k, brown=n, buff=b, chocolate=h, green=r, orange=o, purple=u, white=w, yellow=y

u. population: abundant=a, clustered=c, numerous=n, scattered=s, several=v, solitary=y

v. habitat: grasses=g, leaves=l, meadows=m, paths=p, urban=u, waste=w, woods=d

2.4 The format would as the below mentioned sample



class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attach...
e	x	y	y	t	l	f
e	x	y	y	t	a	f
e	b	s	y	t	a	f
p	x	y	w	t	p	f
e	x	f	n	f	n	f
e	s	f	g	f	n	f
e	f	f	w	f	n	f
p	x	s	n	t	p	f
p	x	y	w	t	p	f
p	x	s	n	t	p	f
e	b	s	y	t	a	f
p	x	y	n	t	p	f
e	b	y	y	t	l	f
e	b	y	w	t	a	f

2.5 Tools Required-

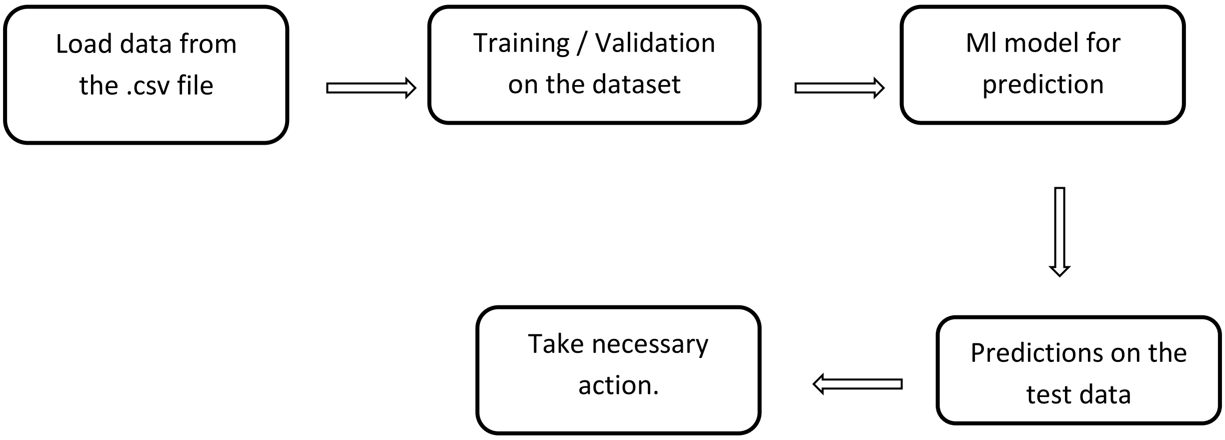
Python Programming language and the frameworks below mentioned are used to build the model.

- Pandas
- Numpy
- Matplotlib
- Sklearn
- XGboost
- Logging
- Flask

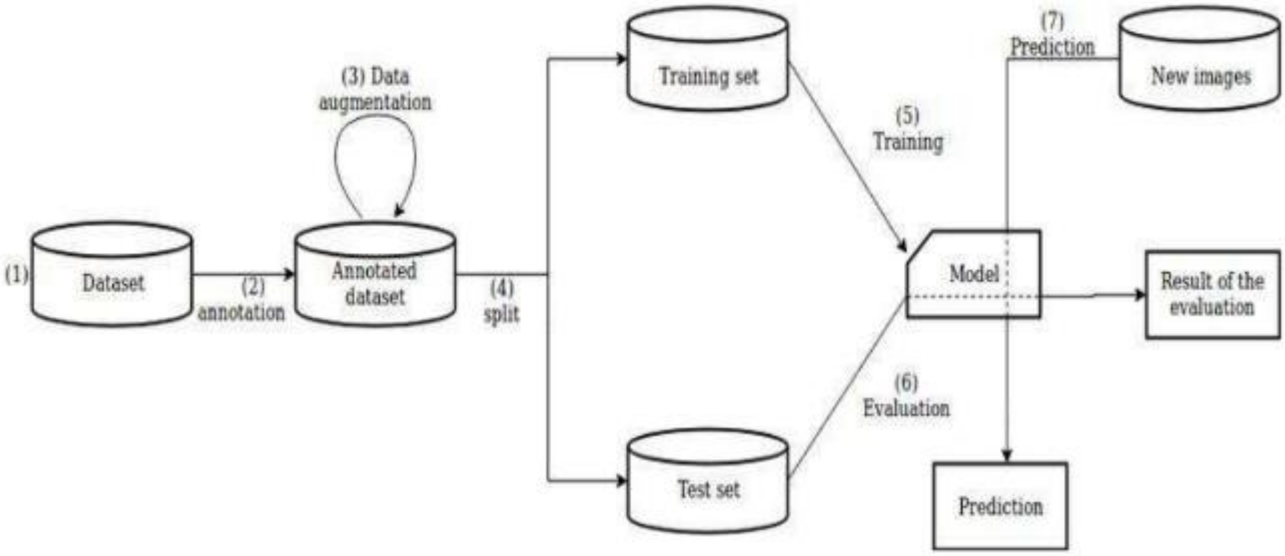
a. VS code is used as the IDE
b. For visualization we user Mat plot library or sea born
c. We use local host for deploying the model
d. HTML/CSS for the front end development
e. Python flask used of the backend development
f. Git is used for version control
g. Docker is used of creation of model that works on any architecture.

3 Design Details

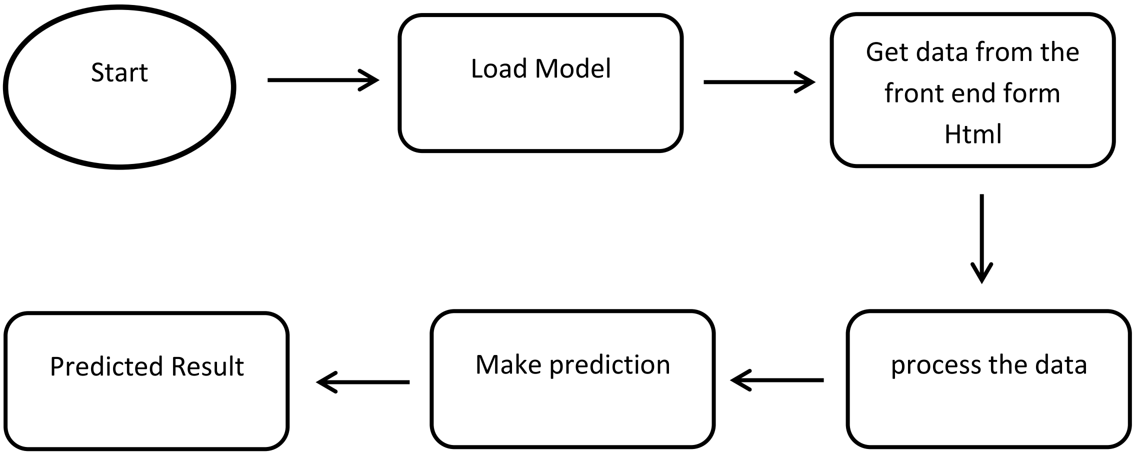
3.1 Process Flow-


Load data from the .csv file
Training / Validation on the dataset
Ml model for prediction
Predictions on the test data
Take necessary action.

3.2 Model Training and Evaluation-



3.3 Deployment Process:



3.4 Event Log-

The system should log every event so the user will know the process that happens on the time.

The system should identify the different logs

The system should note down the logs for further usage

Developer uses this data

3.5 Error Handling-

By using the logs file all errors can be noted and find the solution for them by the developer.

.

4 Performance-

This Mushroom prediction ML model is used to generate the output whether One can have the mushroom or not.

4.1 Reusability-

The code written should be reused without any problem.

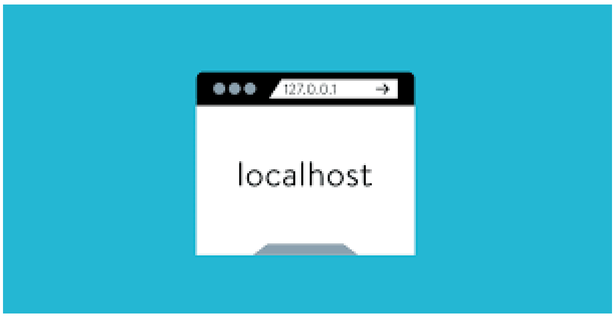
4.2 Application Compatibility-

We are using python as an interface.

4.3 Resource Utilization-

When any task is preformed, it will use all the processing power to do the task assigned.

4.4 Deployment-



5 Conclusion-

This Ml project mushroom classifier which helps user to identify the Mushrooms type (edible or poisonous) so user can used in recipe or not.