# Classification Of Mushrooms

Using Machine Learning Algorithms.

By:(J.S.S.Anvesh, K.Vivek, P.Sandeep)

1

# Index:

- Introduction
- Project Description
- Dataset Features Description
- Data Visualization
- Methods Used In Pre-processing
- Algorithms Performed
- Model Selection

# Introduction:

What is Machine Learning?

- Machine learning, a branch of artificial intelligence, concerns the construction and study of systems that can learn from data.
- "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performances at tasks in T, as measured by P, improves with experience E." Tom M.Mitchell

Why Machine Learning is important?

- The amount of knowledge available about certain tasks might be too large for explicit encoding by humans(e.g., Environments change over time).
- New knowledge about tasks is constantly being discovered by humans. It may be difficult to continuously re-design systems "by hand".

3

# Project Description:

- This project is mainly about classifying a 23 species of gilled mushrooms into 2 categories as Poisonous or Edible.
- Although this dataset was originally contributed to the UCI Machine Learning repository nearly 30 years ago, the dataset was downloaded from www.Kaggle.com
- This dataset includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family Mushroom drawn from The Audubon Society Field Guide to North American Mushrooms (1981).
- Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one.

4

# Dataset Features Description:

- ***Class***: edible=e, poisonous=p
- ***cap-shape:*** bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s
- ***cap-surface:*** fibrous=f, grooves=g, scaly=y, smooth=s
- ***cap-color:*** brown=n, buff=b, cinnamon=c, grey=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y
- ***Bruises:*** bruises=t, no=f
- ***Odour:*** almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m, none=n, pungent=p, spicy=s
- ***gill-attachment:*** attached=a, descending=d, free=f, notched=n
- ***gill-spacing:*** close=c, crowded=w, distant=d
- ***gill-size:*** broad=b, narrow=n
- ***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

5

## Dataset Features Description:

- ***stalk-shape:*** enlarging=e, tapering=t
- ***stalk-root:*** bulbous=b, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r, missing=?
- ***stalk-surface-above-ring:*** fibrous=f, scaly=y, silky=k, smooth=s
- ***stalk-surface-below-ring:*** fibrous=f, scaly=y, silky=k, smooth=s
- ***stalk-color-above-ring:*** brown=n, buff=b, cinnamon=c, grey=g, orange=o, pink=p, red=e, white=w, yellow=y
- ***stalk-color-below-ring:*** brown=n, buff=b, cinnamon=c, grey=g, orange=o, pink=p, red=e, white=w, yellow=y
- ***veil-type:*** partial=p, universal=u
- ***veil-color:*** brown=n, orange=o, white=w, yellow=y
- ***ring-number:*** none=n, one=o, two=t
- ***ring-type:*** cobwebby=c, evanescent=e, flaring=f, large=l, none=n, pendant=p, sheathing=s, zone=z

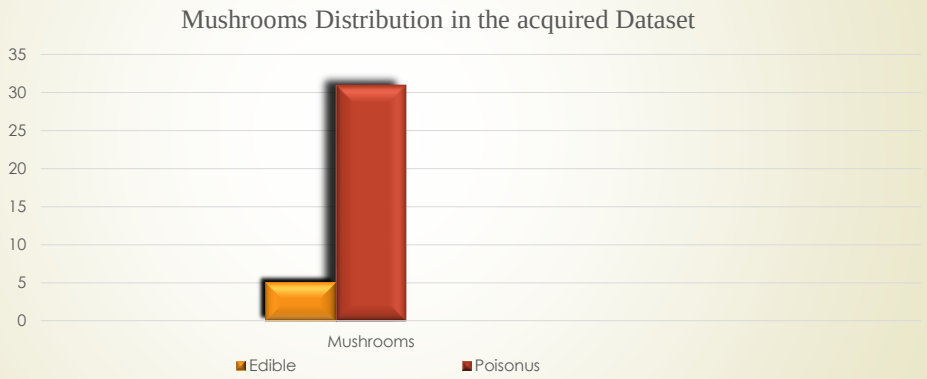
6

## Dataset Features Description:

- ***spore-print-color:*** black=k, brown=n, buff=b, chocolate=h, green=r, orange=o, purple=u, white=w, yellow=y
- ***population:*** abundant=a, clustered=c, numerous=n, scattered=s, several=v, solitary=y
- ***habitat:*** grasses=g, leaves=l, meadows=m, paths=p, urban=u, waste=w, woods=d

7

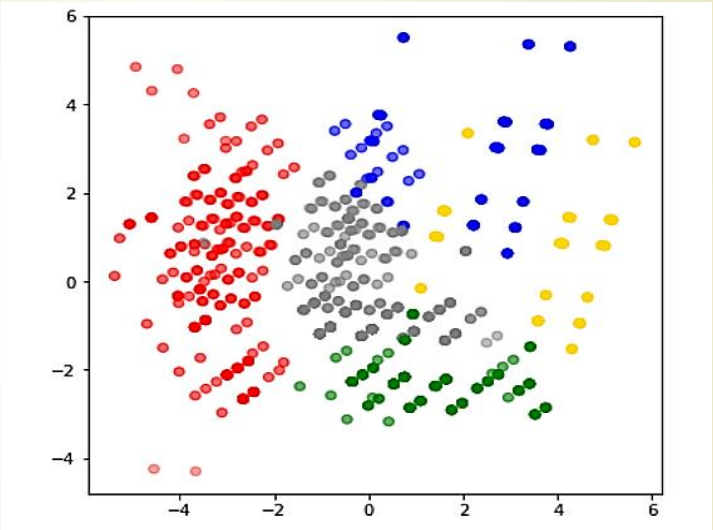
# Data Visualization:




8

# Data Visualization:

- Figure on the right side is scatter plot of the 5 clusters of the color features of the dataset that we made using the kmeans clustering method.



# Data Visualization:

- Figure on the right shows the correlation of the color features of the dataset.

| | cap-color | gill-color | stalk-color-above-ring | stalk-color-below-ring | veil-color | spore-print-color |
|---|---|---|---|---|---|---|
| cap-color | 1.000000 | -0.242099 | 0.046088 | 0.056865 | 0.055906 | -0.105214 |
| gill-color | -0.242099 | 1.000000 | 0.124320 | 0.107444 | 0.065453 | 0.404772 |
| stalk-color-above-ring | 0.046088 | 0.124320 | 1.000000 | 0.503110 | 0.050623 | 0.318965 |
| stalk-color-below-ring | 0.056865 | 0.107444 | 0.503110 | 1.000000 | 0.051065 | 0.279251 |
| veil-color | 0.055906 | 0.065453 | 0.050623 | 0.051065 | 1.000000 | 0.134637 |
| spore-print-color | -0.105214 | 0.404772 | 0.318965 | 0.279251 | 0.134637 | 1.000000 |

# Methods Used In Pre-processing:

- Initially checked for any null values presence in the dataset.

```
data.isnull().sum()
```

```
stalk-shape                   0
stalk-root                 2480
stalk-surface-above-ring      0
```

- As we found some we removed those null values using a command dropna().
- The by using a LableEncoder() we transformed all the categorical data in numeric data. As shown below.

```
le=LabelEncoder()
```

```
data['class']=list(le.fit_transform(data.iloc[:,0]))
data['cap-shape']=list(le.fit_transform(data.iloc[:,1]))
data['cap-surface']=list(le.fit_transform(data.iloc[:,2]))
data['cap-color']=list(le.fit_transform(data.iloc[:,3]))
```

# Algorithms Performed:

- After splitting the data into train and test with a 70% and 30% ratios with respectively.
- Below are the list of classification algorithms that are used for building the required Machine Learning model.
- Logistic Regression-accuracy acquired is 100%.
- K-Nearest Neighbours-accuracy acquired is 100%.
- Support Vector Machine-accuracy acquired is 100%.
- Decision Tree learning- -accuracy acquired is 100%.
- Random Forest -accuracy acquired is 100%.
- Gaussian Naive Bayes -accuracy acquired is 77.8%.
- XG-Boost algorithm -accuracy acquired is 100%.

12

## Model Selection:

- As we can see that almost every algorithm gave an 100% accuracy except for Gaussian Naive Bayes algorithm which gave only 77.8%.
- As the classification of the given dataset is into only two categories (Poisonous or Edible).
- Which is after encoding gets as 0's or 1's.
- So here I prefer to choose Logistic Regression model as it is much suitable for these kind of classification.

```
lr=LogisticRegression()
model=lr.fit(x_train,y_train)
```

```
y_pred=model.predict(x_test)
train_pred=model.predict(x_train)
```

```
print('train=',accuracy_score(train_pred,y_train))
print('test=',accuracy_score(y_pred,y_test))
```

```
train= 1.0
test= 1.0
```