**Code-:**

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

import matplotlib.pyplot as plt

import seaborn as sns

df = pd.read\_csv('../input/mushroom-classification/mushrooms.csv')

df.head()

df.shape

df.isnull().sum()

df.isnull().sum()

plt.figure(figsize = (14,6))

g = sns.catplot(x="cap-shape", hue="cap-color", col="class",

data=df, kind="count",

height=8, aspect=.7);

plt.figure(figsize = (14,6))

g = sns.catplot(x="odor", hue="bruises", col="class",

data=df, kind="count",

height=8, aspect=.7 , palette = 'inferno');

**from** sklearn.preprocessing **import** LabelEncoder

le **=** LabelEncoder()

df **=** df.apply(LabelEncoder().fit\_transform)

df.head()

x = df.drop(['class'] , axis = 1)

y = df['class']

from sklearn.model\_selection import train\_test\_split

x\_train , x\_test , y\_train , y\_test = train\_test\_split(x,y,test\_size = 0.3 , random\_state = 0)

**Decision Tree Model**

**from** sklearn.tree **import** DecisionTreeClassifier

dt **=** DecisionTreeClassifier(random\_state **=** 0 , max\_depth **=** 5)

dt.fit(x\_train , y\_train)

predictions = dt.predict(x\_test)

from sklearn.metrics import accuracy\_score

accuracy\_score(y\_test , predictions)

# ****Random Forest Model****

**from** sklearn.ensemble **import** RandomForestClassifier

rf **=** RandomForestClassifier(max\_depth **=** 5)

rf.fit(x\_train , y\_train)

rf.score(x\_train , y\_train)

predictions **=** rf.predict(x\_test)

rf.score(x\_test , y\_test)

accuracy\_score(y\_test , predictions)