**Table of Contents**

[A prediction algorithm for common mushroom edibility based on ID3 decision tree 2](#_Toc102300989)

[Assignment Question 1: Data Choice 2](#_Toc102300990)

[Assignment Question 2: Background Information 2](#_Toc102300991)

[Assignment Question 3: Data description 3](#_Toc102300992)

[Assignment Question 4: Initial analysis 4](#_Toc102300993)

[Assignment Question 5: GroupBy analysis 5](#_Toc102300994)

[Assignment Question 6: Data visualisation 6](#_Toc102300995)

[Assignment Question 7: Data Mining 8](#_Toc102300996)

[Assignment Question 8: Discussion of findings 9](#_Toc102300997)

[Assignment Question 9: Big Data Management 10](#_Toc102300998)

[References 11](#_Toc102300999)

[Appendices 12](#_Toc102301000)

[Marking Rubrics 17](#_Toc102301001)

**Graphs**

[Figure 6.1 Bar Chart 7](#_Toc102301442)

[Figure 6.2 Bar Chart - 2 8](#_Toc102301443)

[Figure 6.3 Pie Chart 8](#_Toc102301444)

[Figure 6.4 Bar Chart – 3 Frequency List 8](#_Toc102301445)

**Tables**

[Table 3.1 List of Attributes 5](#_Toc102301462)

# A prediction algorithm for common mushroom edibility based on ID3 decision tree

Name: xxxx - Student ID: xxxxx

Course: xxxxx - Professor: xxxxx

## Assignment Question 1: Data Choice

Deep learning has been widely used in plant recognition and classification. Mushrooms are also called large fungi, and wild mushrooms are also called wild fungi. According to reports (State of the World's Fungi 2018) statistics, there are more than 140,000 species in the world are reported. With the continuous improvement of people's living standards, high nutritional value, Wild edible mushrooms with high added value are gradually entering the public's field of vision. Therefore, the classification of mushroom is of great importance in the deep learning area. In this research, we apply ID3 and Cart algorithm to a collection of mushroom records provided by University of California[[1]](#footnote-1)

## Assignment Question 2: Background Information

The collection of data in the mushroom database is partly derived from the book Mushroom records drawn from "The Audubon Society Field Guide to North American Mushroom (1981)". We have 8124 instances in this database. Each piece of data has 22 attributes, all nominally valued.

These data are the more classic and common models in North America selected by the authors of this book after careful selection, so that readers can understand the common mushrooms in North America.

## Assignment Question 3: Data description

In this dataset, there are a total of 8124 instances, of which 4208 are edible, 3916 are unusable parts, and 2480 missing values.

In addition, the data has a total of 22 attributes, and one clear class, which can be categorized as edible and poisonous.

We're going to deduce the classes attribute based on 22 of these attributes to determine if this is edible.

|  |  |  |
| --- | --- | --- |
| 1 | cap-shape: | bell=b,conical=c,convex=x,flat=f,knobbed=k,sunken=s |
| 2 | cap-surface: | fibrous=f,grooves=g,scaly=y,smooth=s |
| 3 | cap-color: | brown=n,buff=b,cinnamon=c,gray=g,green=r,pink=p,purple=u,red=e,white=w,yellow=y |
| 4 | Bruises | bruises=t,no=f |
| 5 | odor | almond=a,anise=l,creosote=c,fishy=y,foul=f,musty=m,none=n,pungent=p,spicy=s |
| 6 | gill-attachme | attached=a,descending=d,free=f,notched=n |
| 7 | gill-spacing | close=c,crowded=w,distant=d |
| 8 | gill-size | broad=b,narrow=n |
| 9 | 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 |
| 10 | stalk-shape | enlarging=e,tapering=t |
| 11 | stalk-root | bulbous=b,club=c,cup=u,equal=e,rhizomorphs=z,rooted=r,missing=? |
| 12 | stalk-surface-above-ring: | fibrous=f,scaly=y,silky=k,smooth=s |
| 13 | stalk-surface-below-ring | fibrous=f,scaly=y,silky=k,smooth=s |
| 14 | stalk-color-a | bove-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o,pink=p,red=e,white=w,yellow=y |
| 15 | stalk-color-b | elow-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o,pink=p,red=e,white=w,yellow=y |
| 16 | veil-type | partial=p,universal=u |
| 17 | veil-color | brown=n,orange=o,white=w,yellow=y |
| 18 | ring-number | none=n,one=o,two=t |
| 19 | ring-type | cobwebby=c,evanescent=e,flaring=f,large=l,none=n,pendant=p,sheathing=s,zone=z |
| 20 | spore-print-c | olor: black=k,brown=n,buff=b,chocolate=h,green=r,orange=o,purple=u,white=w,yellow=y |
| 21 | population | abundant=a,clustered=c,numerous=n,scattered=s,several=v,solitary=y |
| 22 | Habitat | grasses=g,leaves=l,meadows=m,paths=p,urban=u,waste=w,woods=d |

Table 3.1 List of Attributes

## Assignment Question 4: Initial analysis

In this part, we are going to analyze which features to include in our DataFrame and how to deal with missing values (if they exist).

By calling the functions of pandas, we can quickly have an overview of the database.

# Q3 initial analysis.

#  how many instances does the dataset contain

print("The dataset contains",len(data),"instances")

# how many attributes there are in the dataset

print("The dataset contains",len(data.columns),"attributes")

# decide if there is missing value

print("There are",data.isnull().sum().sum(),"missing values")

#  how many attributes there are in the dataset, their names, and include which is the class attribute.

**for** col **in** data.columns:

    print("The [",col,"]is",data[col].dtype)

    print("The dataset contains",data[col].value\_counts(),"instances in each class")

    print("-----------------------------------------------------------------------------------")

# preprocess the data

processed\_data = data.drop(['classes'],axis=1)

## Assignment Question 5: GroupBy analysis

Here, we are going to group each attribute of the data, for example, as we mentioned at the beginning, the classes of the mushroom dataset can be simply divided into edible and poisonous.

Here, we take cap-shape as an example to perform group by analysis.

# Q4 groupby analysis

# conducting groupby analysis

print("the cap-shape has "+str(processed\_data['cap-shape'].nunique())+" unique values")

# list the unique value

print(processed\_data['cap-shape'].unique())

# a list of the unique values

cap\_shape\_x=(processed\_data['cap-shape']=='x')

cap\_shape\_b=(processed\_data['cap-shape']=='b')

cap\_shape\_s=(processed\_data['cap-shape']=='s')

cap\_shape\_f=(processed\_data['cap-shape']=='f')

cap\_shape\_k=(processed\_data['cap-shape']=='k')

cap\_shape\_c=(processed\_data['cap-shape']=='c')

print("the cap-shape has "+str(cap\_shape\_x.sum())+" x")

print("the cap-shape has "+str(cap\_shape\_b.sum())+" b")

print("the cap-shape has "+str(cap\_shape\_s.sum())+" s")

print("the cap-shape has "+str(cap\_shape\_f.sum())+" f")

print("the cap-shape has "+str(cap\_shape\_k.sum())+" k")

print("the cap-shape has "+str(cap\_shape\_c.sum())+" c")

# groupby analysis

cap\_shape\_data=processed\_data.groupby('cap-shape')

cap\_shape\_data.get\_group('x')

cap\_shape\_data.get\_group('x').shape

print(type(cap\_shape\_data))

print(type(cap\_shape\_data.get\_group('x')))

print(len(cap\_shape\_data))

print(len(cap\_shape\_data.get\_group('x')))

## Assignment Question 6: Data visualisation

Bar Chart Analysis:

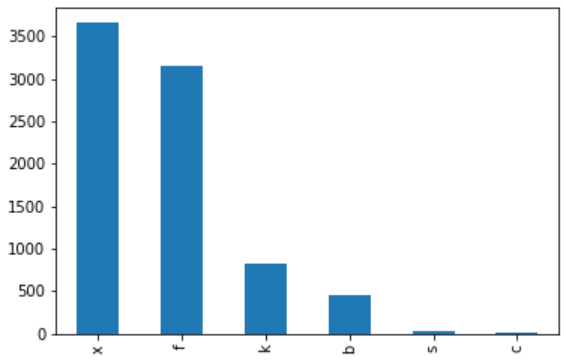


Figure 6.1 Bar Chart

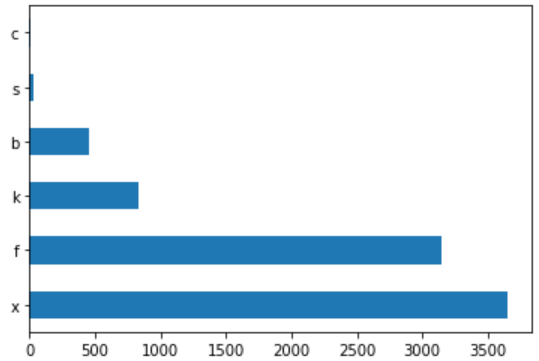


Figure 6.2 Bar Chart - 2

Pie Chart Analysis:

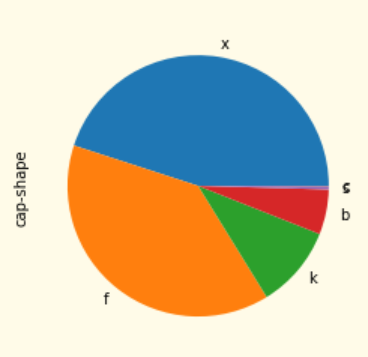


Figure 6.3 Pie Chart

Frequency analysis:

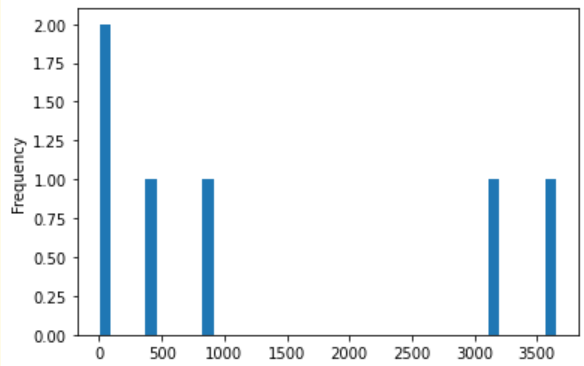


Figure 6.4 Bar Chart – 3 Frequency List

## Assignment Question 7: Data Mining

Here, we choose two common decision tree algorithms, ID3 and Cart, to fit and predict the data set, and finally compare and analyze the two algorithms according to the accuracy.

# Q7 Data mining

# split the data

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

train\_lr, test\_lr = train\_test\_split(data, test\_size=0.33)

# feature and target

data\_features = ['cap-shape','cap-surface','cap-color','bruises?','odor','gill-attract','gill-spacing','gill-size','gill-color','stalk-shape','stalk-root','stalk-surface-above-ring','stalk-surface-below-ring','stalk-color-above-ring','stalk-color-below-ring','veil-type','veil-color','ring-number','ring-type','spore-print-color','population','habitat']

data\_target = ['classes']

# ID3 algorithm

**from** sklearn.feature\_extraction **import** DictVectorizer

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

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

**from** sklearn.metrics **import** accuracy\_score

vec=DictVectorizer(sparse=False)

X\_train=vec.fit\_transform(train\_lr[data\_features].to\_dict(orient='records'))

y\_train=train\_lr[data\_target].values

X\_test=vec.transform(test\_lr[data\_features].to\_dict(orient='records'))

y\_test=test\_lr[data\_target].values

dtc\_model = DecisionTreeClassifier()

dtc\_model.fit(X\_train,y\_train)

y\_pred = dtc\_model.predict(X\_test)

print("the accuracy of the model is: "+str(accuracy\_score(y\_test,y\_pred)))

# cart algorithm

rf\_model = RandomForestClassifier(n\_estimators=100)

rf\_model.fit(X\_train,y\_train)

y\_pred = rf\_model.predict(X\_test)

print("the accuracy of the model is: "+str(accuracy\_score(y\_test,y\_pred)))

## Assignment Question 8: Discussion of findings

The two algorithms proposed in this paper have high accuracy for the classification of mushrooms, which may be because the mushrooms in the datasets provided in the books are carefully selected and have strong characteristics. Therefore, feature-based classification methods like ID3, the effect of C4.5 and Cart will be more obvious.

The learning process of decision tree consists of three steps:

a) Feature selection. Different features and prediction targets have different strengths of correlation, and selecting the feature with the strongest correlation can effectively improve the prediction effect.

b) Node splits. The training set will be divided according to the node rules in the decision tree. If node A cannot give a satisfactory classification result, it will choose to split into 2 or more nodes. So, what is the division based on? Node A will use entropy to determine which feature split is optimal to use.

c) Pruning. Unrestricted splitting of decision trees is prone to over-fitting. Pruning can alleviate over-fitting to a certain extent and improve generalization ability.

The feature selection contained in this process can also be the reason why the accuracy is high.

## Assignment Question 9: Big Data Management

The data set in this paper can be further expanded, especially for mushrooms with different characteristics and different scenes, and more data augmentation methods can be used to expand the data set to further expand the scope of model use and improve model performance.

The experiments in this paper use the GPU of the online platform for training, and the training equipment and methods can be improved in the future to improve the efficiency of model training.

More parameters can also be optimized in the experiment to obtain a model with higher accuracy, thus forming a mushroom image classifier that can be put into practical application. The model in this paper, combined with mobile devices and embedded development, can realize mushroom classification and identification in the wild, which has important significance and application value.

**References**

**Appendix1**

**Appendices**

**import** keras

**import** numpy

**import** sklearn

**import** pyspark

**import** tensorflow

**import** pandas **as** pd

**import** matplotlib.pyplot **as** plt

# Q1 data choice

# Census-Income Database

# https://archive.ics.uci.edu/ml/datasets/Census-Income+%28KDD%29

data = pd.read\_csv('./agaricus-lepiota.data',sep=',')

data.columns = ['classes','cap-shape','cap-surface','cap-color','bruises?','odor','gill-attract','gill-spacing','gill-size','gill-color','stalk-shape','stalk-root','stalk-surface-above-ring','stalk-surface-below-ring','stalk-color-above-ring','stalk-color-below-ring','veil-type','veil-color','ring-number','ring-type','spore-print-color','population','habitat']

print(data.head)

print(data.shape)

print(data.columns)

print(data.describe())

# Q2 data description

# From Audobon Society Field Guide; mushrooms described in terms of physical characteristics; classification: poisonous or edible. extracted from the 1994 and 1995 Current Census Surveys conducted by the U.S. Census Bureau. The data contains 41 demographic and employment-related variables.

# Mushroom records taken from the Audubon Society Field Guide to North American Mushrooms

# The dataset includes descriptions of hypothetical samples corresponding to 23 species of spiny mushrooms in the Agaricus and Lepiota families

# Q3 initial analysis.

#  how many instances does the dataset contain

print("The dataset contains",len(data),"instances")

# how many attributes there are in the dataset

print("The dataset contains",len(data.columns),"attributes")

# decide if there is missing value

print("There are",data.isnull().sum().sum(),"missing values")

#  how many attributes there are in the dataset, their names, and include which is the class attribute.

**for** col **in** data.columns:

    print("The [",col,"]is",data[col].dtype)

    print("The dataset contains",data[col].value\_counts(),"instances in each class")

    print("-----------------------------------------------------------------------------------")

# preprocess the data

processed\_data = data.drop(['classes'],axis=1)

# Q4 groupby analysis

# conducting groupby analysis

print("the cap-shape has "+str(processed\_data['cap-shape'].nunique())+" unique values")

# list the unique value

print(processed\_data['cap-shape'].unique())

# a list of the unique values

cap\_shape\_x=(processed\_data['cap-shape']=='x')

cap\_shape\_b=(processed\_data['cap-shape']=='b')

cap\_shape\_s=(processed\_data['cap-shape']=='s')

cap\_shape\_f=(processed\_data['cap-shape']=='f')

cap\_shape\_k=(processed\_data['cap-shape']=='k')

cap\_shape\_c=(processed\_data['cap-shape']=='c')

print("the cap-shape has "+str(cap\_shape\_x.sum())+" x")

print("the cap-shape has "+str(cap\_shape\_b.sum())+" b")

print("the cap-shape has "+str(cap\_shape\_s.sum())+" s")

print("the cap-shape has "+str(cap\_shape\_f.sum())+" f")

print("the cap-shape has "+str(cap\_shape\_k.sum())+" k")

print("the cap-shape has "+str(cap\_shape\_c.sum())+" c")

# groupby analysis

cap\_shape\_data=processed\_data.groupby('cap-shape')

cap\_shape\_data.get\_group('x')

cap\_shape\_data.get\_group('x').shape

print(type(cap\_shape\_data))

print(type(cap\_shape\_data.get\_group('x')))

print(len(cap\_shape\_data))

print(len(cap\_shape\_data.get\_group('x')))

# Q5 data representation

processed\_data['cap-shape'].value\_counts().plot(kind='bar')

processed\_data['cap-shape'].value\_counts().plot(kind='barh')

processed\_data['cap-shape'].value\_counts().plot(kind='pie')

processed\_data['cap-shape'].value\_counts().plot(kind='hist',bins=40)

# Q7 Data mining

# split the data

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

train\_lr, test\_lr = train\_test\_split(data, test\_size=0.33)

# feature and target

data\_features = ['cap-shape','cap-surface','cap-color','bruises?','odor','gill-attract','gill-spacing','gill-size','gill-color','stalk-shape','stalk-root','stalk-surface-above-ring','stalk-surface-below-ring','stalk-color-above-ring','stalk-color-below-ring','veil-type','veil-color','ring-number','ring-type','spore-print-color','population','habitat']

data\_target = ['classes']

# ID3 algorithm

**from** sklearn.feature\_extraction **import** DictVectorizer

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

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

**from** sklearn.metrics **import** accuracy\_score

vec=DictVectorizer(sparse=False)

X\_train=vec.fit\_transform(train\_lr[data\_features].to\_dict(orient='records'))

y\_train=train\_lr[data\_target].values

X\_test=vec.transform(test\_lr[data\_features].to\_dict(orient='records'))

y\_test=test\_lr[data\_target].values

dtc\_model = DecisionTreeClassifier()

dtc\_model.fit(X\_train,y\_train)

y\_pred = dtc\_model.predict(X\_test)

print("the accuracy of the model is: "+str(accuracy\_score(y\_test,y\_pred)))

# cart algorithm

rf\_model = RandomForestClassifier(n\_estimators=100)

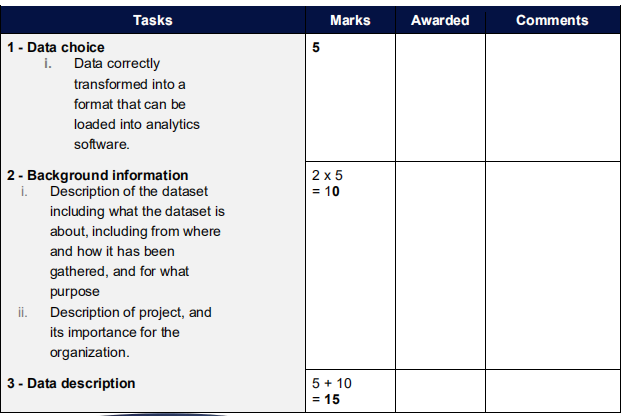
rf\_model.fit(X\_train,y\_train)

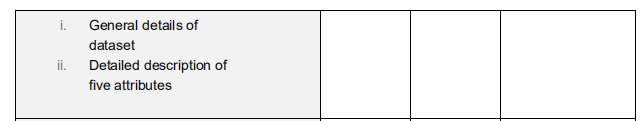
y\_pred = rf\_model.predict(X\_test)

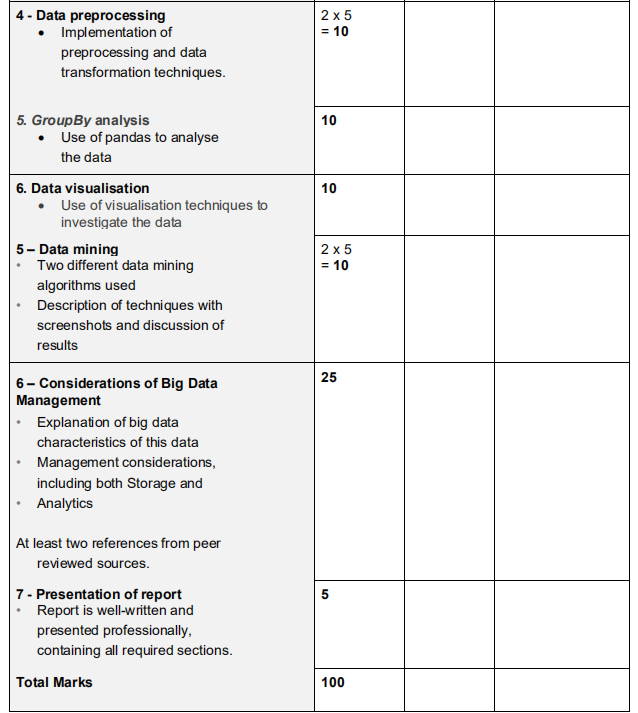
print("the accuracy of the model is: "+str(accuracy\_score(y\_test,y\_pred)))

**Appendix2**

**Marking Rubrics**









1. https://archive.ics.uci.edu/ml/datasets/mushroom [↑](#footnote-ref-1)