Sagar Limbu

Analysis of Mushroom Dataset

- Given: 23 Species of GILLED mushrooms; AGARICUS and LEPIOTA Family Mushroom
- Objective: Detect the class of mushrooms; edible, definitely poisonous or unknown edibility and not recommended
- Source: Kaggle

Possible analysis

- Which Machine Learning Model can accurately classify the mushroom dataset?
- Performing ML techniques; Pandas, Numpy, sckikit-learn metrics, graphviz, Accuracy_Score, ROC curve, and Cross Validation
- Which **features** are most indicative for a poisonous or edible mushroom?

Reading the mushroom csv file

After importing the dataset, we cleaned, aggregated and extracted required features that would best fit the Machine Learning model for accurate analysis.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns ; sns.set(style="ticks", color_codes=True)
```

```
df_mush= pd.read_csv("mushrooms.csv")
#df_mush_info= pd.read_csv("columns_definitions.csv")
```

```
df_mush.head(5)
```

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring-number	ring-type	spore-print-color	population
0	p	x	s	n	t	p	f	c	n	k	...	s	w	w	p	w	o	p	k	s
1	e	x	s	y	t	a	f	c	b	k	...	s	w	w	p	w	o	p	n	n
2	e	b	s	w	t	l	f	c	b	n	...	s	w	w	p	w	o	p	n	n
3	p	x	y	w	t	p	f	c	n	n	...	s	w	w	p	w	o	p	k	s
4	e	x	s	g	f	n	f	w	b	k	...	s	w	w	p	w	o	e	n	a

FINDINGS:

- We found several missing values ‘?’ on the data set
- We also detected that the column ‘viel-type’ has only one values in the entire data set

	Unnamed: 0	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring-number	ring-type	spore-print-color	popula
0	0	x	y	y	f	f	f	c	b	h	...	k	n	b	p	w	o	l	h	
1	1	b	s	y	t	l	f	c	b	k	...	s	w	w	p	w	o	p	n	
2	2	x	f	n	t	n	f	c	b	u	...	s	p	w	p	w	o	p	n	
3	3	k	s	n	f	y	f	c	n	b	...	s	w	p	p	w	o	e	w	
4	4	f	s	e	f	f	f	c	n	b	...	k	w	p	p	w	o	e	w	

5 rows × 23 columns

```
df_sub['veil-type'].unique()
```

```
array(['p'], dtype=object)
```

Why drop column ‘viel-type’ from the dataset?

The column ‘viel-type’ has only one value ‘p’ on the **entire** dataset. Additionally, the following findings suggests more evidence to drop the column.

Veils

Some mushrooms have membranes referred to as veils that cover and protect the immature mushroom or the immature lamellae. There are two types of veils.

The membrane protecting the immature lamellae is called the partial veil and the membrane covering the immature mushroom is called the universal veil. As the mushroom matures these membranes break, sometimes leaving remnants on the stipe or the pileus

Features of the mushroom:

Further analysis suggested that we have the following features to make analysis on the classification of the mushroom.

```
# Features

features= X.columns.to_list()
features
```

```
['cap-shape',
 'cap-surface',
 'cap-color',
 'bruises',
 'odor',
 'gill-attachment',
 '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-color',
 'ring-number',
 'ring-type',
 'spore-print-color',
 'population',
 'habitat']
```

USE LABEL ENCODER for X FEATURES

```
label_en= LabelEncoder()

for i in X.columns:
    X[i]= label_en.fit_transform(X[i])
```

Transforming the values of data set

Here, using Label Encoder function we have transformed each categorical value to numerical value. In other words, assigning each alphabetical character to numerical value. For Example,

- mushroom gill-color: ‘k’ -> ‘4’
- mushroom gill-color: ‘n’ -> ‘5’

Replacing the Missing Values

We also detected the missing values on the dataset (‘?’) on the column ‘stalk-root’. We replaced the value with char ‘b’ by taking the average value of the same column.

MISSING Values from column 'Stalk- ROOT'

```
df_new['stalk-root'].unique()
```

```
array(['e', 'c', 'b', 'r', '?'], dtype=object)
```

```
# replace With MODE value from the olumn
df_new['stalk-root'] = df_new['stalk-root'].str.replace('?','b')
df_new['stalk-root'].unique()
```

```
array(['e', 'c', 'b', 'r'], dtype=object)
```

Analysis and applying ML models:

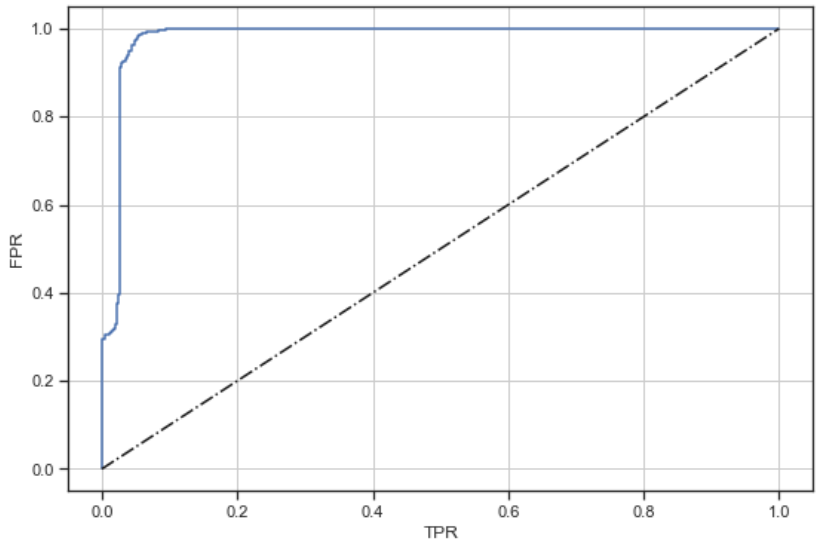
After cleaning the dataset and performing data aggregation, we applied different ML models to get the possible accurate prediction for the classification of either ‘Edible’ or ‘Poisonous’ mushroom based on the features provided.

We also performed test with accuracy score and applied confusion matrix on our findings.

USING LOGISTIC REGRESSION,, ROC CURVE

```
model =LR()

xtrain, xtest, ytrain, ytest= train_test_split(model_df, Y, test_size=0.2)

model.fit(xtrain, ytrain)
ypredict= model.predict(xtest)

acc_1= accuracy_score(ytest, ypredict)
prob= model.predict_proba(xtest)[:,1]
FPR, TPR, Thresholds= roc_curve(ytest, prob)

plt.plot(FPR, TPR)

plt.gcf().set_size_inches(9,6)
# 45 degree line
plt.plot([0,1], [0,1], c='k', ls='dashdot', label="45 Degree Line")

plt.ylabel('FPR', fontsize= '12')
plt.xlabel('TPR', fontsize= '12')
plt.grid()
```

After using the Logistic Regression along the ROC curve, we found some interesting output.




```
print("Accuracy_SCORE :",acc_1)
```

```
Accuracy_SCORE : 0.9533123028391167
```

```
print("AUC: ", auc(FPR, TPR))
```

```
AUC:  0.9797427293064878
```

CONFUSION MATRIX with LogR

```
ypredict_1= model.predict(model_df)

# with target Feature
confusion= confusion_matrix(Y, ypredict_1)
confusion
```

```
array([[3935,  161],
       [ 172, 3653]], dtype=int64)
```

Applying Decission Tree Classification

Further, we used Decision Tree Classification model to analyze the data set. Surprisingly, DT model also showed the results that the features of mushroom was highly accurate on classification of mushrooms.

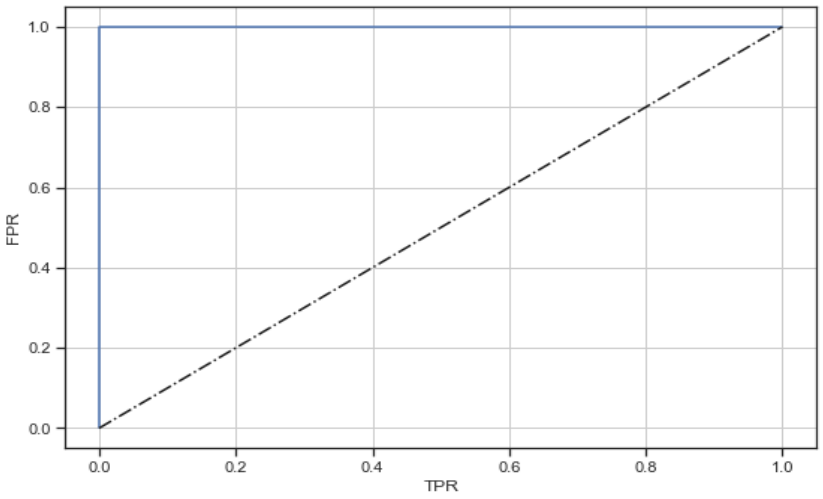
DECISSION TREE CLASSIFIER

```
from sklearn.tree import DecisionTreeClassifier as DT
```

```
xtrain, xtest, ytrain, ytest= train_test_split(model_df, Y)

model_DT= DT( max_depth= 4, criterion='gini')
```

```
model_DT.fit(xtrain, ytrain)
ypredict_dt= model_DT.predict(xtest)


model_DT.fit(xtrain, ytrain)
ypredict_dt= model_DT.predict(xtest)

acc_dt= accuracy_score(ytest, ypredict_dt)
prob_1= model_DT.predict_proba(xtest)[:,1]
FPR, TPR, Thresholds= roc_curve(ytest, prob_1)

auc_DT= auc(FPR,TPR)
plt.plot(FPR, TPR)

plt.gcf().set_size_inches(9,6)
# 45 degree line
plt.plot([0,1], [0,1], c='k', ls='dashdot', label="45 Degree Line")

plt.ylabel('FPR', fontsize= '12')
plt.xlabel('TPR', fontsize= '12')
plt.grid()
```

Accuracy score and ROC Curve




```
: print("Decission Tree Acc: ", acc_dt)
```

```
Decission Tree Acc:  1.0
```

```
: print("Decission Tree AUC: ", auc_DT)
```

```
Decission Tree AUC:  1.0
```

Visualizing the Decision Tree:

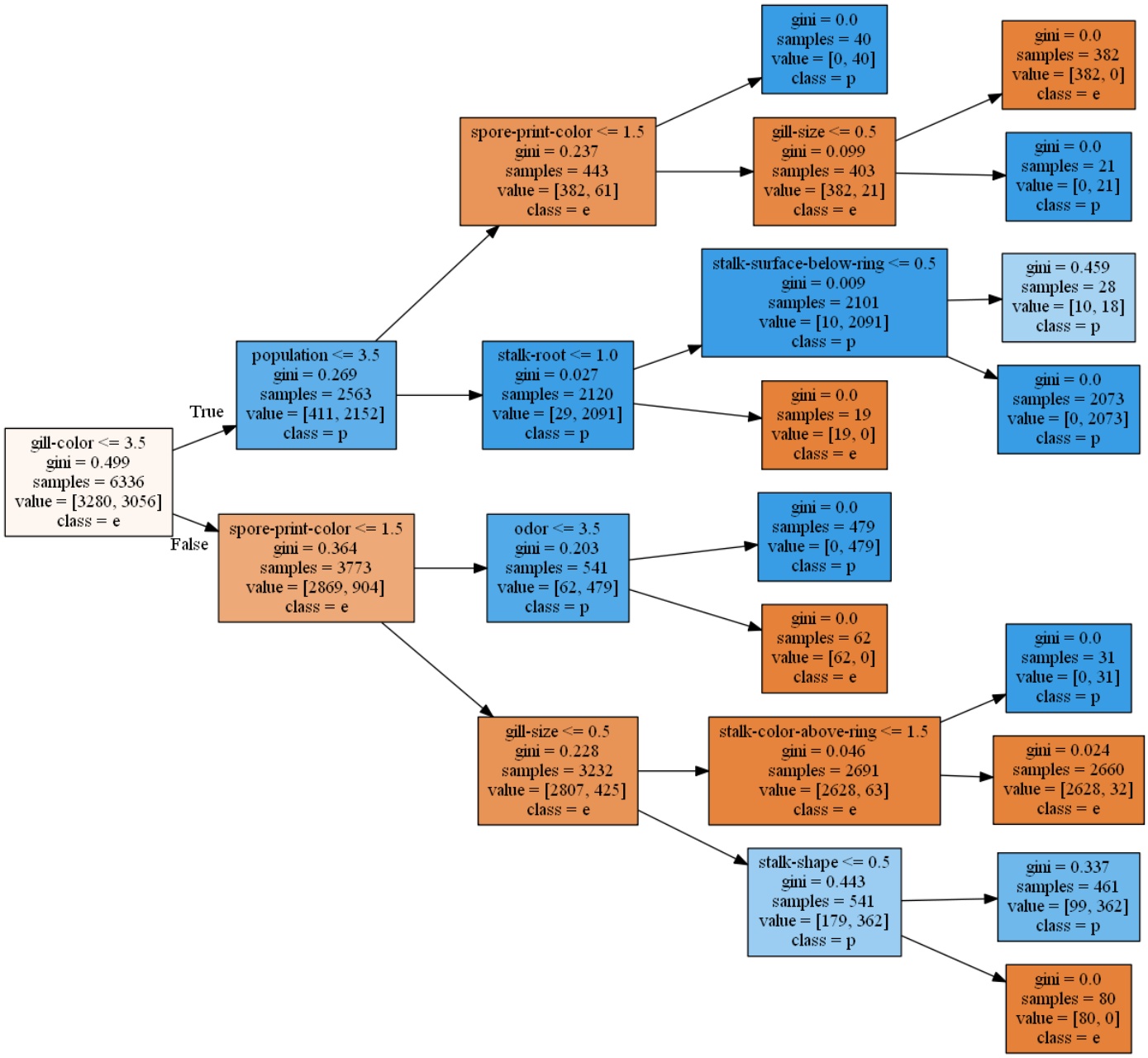
To make clear presentation and analysis on the Decision Tree ML model, we imported the required packages and drew a graph. We can see that each individual feature has been sorted and represented as the most important features and less important.

```
import graphviz
import pydotplus
from sklearn.tree import export_graphviz
from IPython.display import Image
```

```

```

```
dot= export_graphviz(model_DT,
                     rotate= True,
                     filled= True,
                     feature_names= features,
                     class_names= [str(i) for i in classes]
                     )
```

```
g= pydotplus.graph_from_dot_data(dot)
Image(g.create_png())
```


gill-color <= 3.5
gini = 0.499
samples = 6336
value = [3280, 3056]
class = e
True
False
population <= 3.5
gini = 0.269
samples = 2563
value = [411, 2152]
class = p
spore-print-color <= 1.5
gini = 0.364
samples = 3773
value = [2869, 904]
class = e
spore-print-color <= 1.5
gini = 0.237
samples = 443
value = [382, 61]
class = e
stalk-root <= 1.0
gini = 0.027
samples = 2120
value = [29, 2091]
class = p
odor <= 3.5
gini = 0.203
samples = 541
value = [62, 479]
class = p
gill-size <= 0.5
gini = 0.228
samples = 3232
value = [2807, 425]
class = e
gini = 0.0
samples = 40
value = [0, 40]
class = p
gill-size <= 0.5
gini = 0.099
samples = 403
value = [382, 21]
class = e
stalk-surface-below-ring <= 0.5
gini = 0.009
samples = 2101
value = [10, 2091]
class = p
gini = 0.0
samples = 19
value = [19, 0]
class = e
gini = 0.0
samples = 479
value = [0, 479]
class = p
gini = 0.0
samples = 62
value = [62, 0]
class = e
stalk-color-above-ring <= 1.5
gini = 0.046
samples = 2691
value = [2628, 63]
class = e
stalk-shape <= 0.5
gini = 0.443
samples = 541
value = [179, 362]
class = p
gini = 0.0
samples = 382
value = [382, 0]
class = e
gini = 0.0
samples = 21
value = [0, 21]
class = p
gini = 0.459
samples = 28
value = [10, 18]
class = p
gini = 0.0
samples = 2073
value = [0, 2073]
class = p
gini = 0.0
samples = 31
value = [0, 31]
class = p
gini = 0.024
samples = 2660
value = [2628, 32]
class = e
gini = 0.337
samples = 461
value = [99, 362]
class = p
gini = 0.0
samples = 80
value = [80, 0]
class = e

Using Cross-Validation Score to check the analysis

To check if the situation is **overfitting**, we also performed Cross-Validation Score to check our analysis on Decision Tree model.

```
from sklearn.model_selection import cross_val_score as cv
```

```
logR_cv= cv(DT(max_depth=4, criterion='gini'), model_df, Y)
print("cross validation score: ", logR_cv)
print("AVERAGE cross validation score for DT: ", sum(logR_cv)/len(logR_cv))
```

```
cross validation score for Logistic_Regression:  [0.84227129 0.98674242 0.94065657 1.         0.74621212]
AVERAGE cross validation score for DT:  0.9031764808973011
```

Conclusion:

```
print("total poisonous: ", c)
print("total edible: ", d)
```

```
total poisonous:  89
total edible:  114
```

```
total poisonous:  91
total edible:  112
```

We performed several data cleaning technique steps and data aggregation to extract the important features from the given dataset.

After performing two different ML models on the mushroom dataset our Finding suggest that, the feature ‘gill-color’ is the most important of all. This feature ‘gill-color’ can highly predict than the rest features whether the mushroom is ‘Edible’ or ‘Poisonous’.

References:

- https://cooking.stackexchange.com/questions/65575/what-is-this-part-of-the-mushroom-is-it-safe-to-eat