

```
In [ ]: # This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load
```

```
# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
df = pd.read_csv('../input/mush
```

```
df.head()
```

class C

0	p	x	s	n	t	p	f	c	n	k ...	s	w	w	p	w	o	p
1	e	x	s	y	t	a	f	c	b	k ...	s	w	w	p	w	o	p
2	e	b	s	w	t	l	f	c	b	n ...	s	w	w	p	w	o	p
3	p	x	y	w	t	p	f	c	n	n ...	s	w	w	p	w	o	p
4	e	x	s	g	f	n	f	w	b	k ...	s	w	w	p	w	o	e

5 rows × 23 columns

```
df.info()
```

\_\_\_\_\_

2	cap-surface	8124	non-null	object
3	cap-color	8124	non-null	object
4	bruises	8124	non-null	object
5	odor	8124	non-null	object
6	gill-attachment	8124	non-null	object
7	gill-spacing	8124	non-null	object
8	gill-size	8124	non-null	object
9	gill-color	8124	non-null	object
10	stalk-shape	8124	non-null	object
11	stalk-root	8124	non-null	object
12	stalk-surface-above-ring	8124	non-null	object
13	stalk-surface-below-ring	8124	non-null	object
14	stalk-color-above-ring	8124	non-null	object
15	stalk-color-below-ring	8124	non-null	object
16	veil-type	8124	non-null	object
17	veil-color	8124	non-null	object
18	ring-number	8124	non-null	object
19	ring-type	8124	non-null	object
20	spore-print-color	8124	non-null	object
21	population	8124	non-null	object
22	habitat	8124	non-null	object

dtypes: object (23)

```
memory usage: 1.4+ MB
```

```
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-type
veil-color
ring-number
ring-type
spore-print-color
population
habitat
dtype: int64
```

```
<class 'pandas.co
```

```

# Column Non-Null Count Dtype
---  ---
0 class 8124 non-null object
1 cap-shape 8124 non-null object
2 cap-surface 8124 non-null object
3 cap-color 8124 non-null object
4 bruises 8124 non-null object
5 odor 8124 non-null object
6 gill-attachment 8124 non-null object
7 gill-spacing 8124 non-null object
8 gill-size 8124 non-null object
9 gill-color 8124 non-null object
10 stalk-shape 8124 non-null object
11 stalk-root 8124 non-null object
12 stalk-surface-above-ring 8124 non-null object
13 stalk-surface-below-ring 8124 non-null object
14 stalk-color-above-ring 8124 non-null object
15 stalk-color-below-ring 8124 non-null object
16 veil-type 8124 non-null object
17 veil-color 8124 non-null object
18 ring-number 8124 non-null object
19 ring-type 8124 non-null object
20 spore-print-color 8124 non-null object
21 population 8124 non-null object
22 habitat 8124 non-null object

dtypes: object(23)
memory usage: 1.5+ MB

pd.get_dummies(df)

```

class\_e class\_p cha

<b>3</b>	0	1	0	0	0	0	0	0	1	0	0 ...	1	0
<b>4</b>	1	0	0	0	0	0	0	0	1	0	0 ...	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
<b>8119</b>	1	0	0	0	0	0	1	0	0	0	0 ...	0	0
<b>8120</b>	1	0	0	0	0	0	0	0	1	0	0 ...	0	1
<b>8121</b>	1	0	0	0	0	1	0	0	0	0	0 ...	0	0
<b>8122</b>	0	1	0	0	0	0	1	0	0	0	0 ...	0	1
<b>8123</b>	1	0	0	0	0	0	0	0	1	0	0 ...	0	0

```
In [9]: x = pd.get_dummies(df.drop('class',axis=1),drop_first=True)
```

```
In [14]: y.head()
```

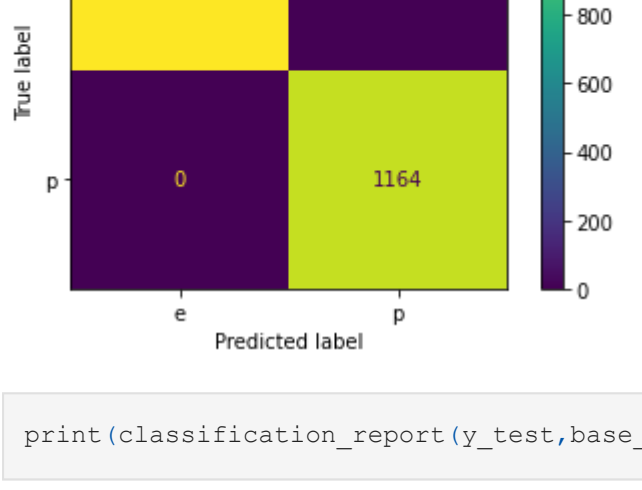
```
0    p
1    e
2    e
3    p
4    e
Name: class, dtype: object
```

```
model = DecisionTreeClassifier()
model.fit(X_train, y_train)
base_pred = model.predict(X_test)
```

```
Out[18]: array([[ 0, 1164]])
```

Year	Number of people
2010	1200
2015	1250

e	12/4	0	
---	------	---	--



e	1.00	1.00	1.00
---	------	------	------

p	1.00	1.00	1.00	1164
accuracy			1.00	2438
macro avg	1.00	1.00	1.00	2438
weighted avg	1.00	1.00	1.00	2438
model.feature importances				

```
Out[21]: array([0.00000000e+00, 0.00000000e+00, 3.10610611e-01,
```

```
0.00000000e+00, 1.38706794e-03, 0.00000000e+00, 0.00000000e+00,  
1.17443865e-02, 0.00000000e+00, 6.29299472e-01, 0.00000000e+00,  
0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 2.18493923e-03,  
2.01574610e-05, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,  
0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,  
0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,  
9.71018688e-03, 0.00000000e+00, 1.80465344e-01, 0.00000000e+00,  
7.25167313e-02, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,  
0.00000000e+00, 1.58390534e-02, 1.53697829e-02, 0.00000000e+00,  
0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,  
0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,  
0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,  
0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,  
0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,  
0.00000000e+00, 0.00000000e+00, 6.09218104e-03, 0.00000000e+00,  
0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,  
0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,  
2.88786035e-02, 2.43856790e-02, 0.00000000e+00, 0.00000000e+00,  
0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,  
0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,  
0.00000000e+00, 0.00000000e+00, 0.00000000e+00}}  
  
pd.DataFrame(index=X.columns,data=model.feature_importances_,columns=['Feature Importance'])
```

Output:

Feature Importance

cap-shape_s	0.0
cap-shape_x	0.0
...	...
habitat_l	0.0
habitat_m	0.0
habitat_p	0.0
habitat_u	0.0
habitat_w	0.0

95 rows x 1 columns

```
from sklearn.tree import
```

```
plt.figure(figsize=(10, 10))
```

