

```
In [87]: import numpy as np
import pandas as pd
from functools import reduce

from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, f1_score
from sklearn.preprocessing import OneHotEncoder
```

```
In [88]: df = pd.read_csv('mushrooms.csv')
df
```

Out[88]:

| | 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 | 1 |
|------|-------|---------------|-----------------|---------------|---------|------|---------------------|------------------|---------------|----------------|-----|--------------------------------------|------------------------------------|------------------------------------|---|
| 0 | p | x | s | n | t | p | f | c | n | k | ... | s | w | w | |
| 1 | e | x | s | y | t | a | f | c | b | k | ... | s | w | w | |
| 2 | e | b | s | w | t | l | f | c | b | n | ... | s | w | w | |
| 3 | p | x | y | w | t | p | f | c | n | n | ... | s | w | w | |
| 4 | e | x | s | g | f | n | f | w | b | k | ... | s | w | w | |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| 8119 | e | k | s | n | f | n | a | c | b | y | ... | s | o | o | |
| 8120 | e | x | s | n | f | n | a | c | b | y | ... | s | o | o | |
| 8121 | e | f | s | n | f | n | a | c | b | n | ... | s | o | o | |
| 8122 | p | k | y | n | f | y | f | c | n | b | ... | k | w | w | |
| 8123 | e | x | s | n | f | n | a | c | b | y | ... | s | o | o | |

8124 rows × 23 columns



```
In [89]: x = df.iloc[:, 1:]
ohe = OneHotEncoder()
arr = ohe.fit_transform(x).toarray()
labels = [i.tolist() for i in ohe.categories_]
labels = reduce(lambda x, y: x+y, labels)

x = pd.DataFrame(arr, columns=labels)
x
```

Out[89]:

| | b | c | f | k | s | x | f | g | s | y | ... | s | v | y | d | g | l | m | p | u | w |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | ... | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 |
| 1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 |
| 3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | ... | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 |
| 4 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 8119 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 8120 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | ... | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 8121 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 8122 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | ... | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 8123 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |

8124 rows × 117 columns

```
In [90]: y = df.iloc[:,0]
```

```
In [91]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
```

```
In [92]: x_train.shape, x_test.shape, y_train.shape, y_test.shape
```

```
Out[92]: ((6499, 117), (1625, 117), (6499,), (1625,))
```

```
In [93]: ada = AdaBoostClassifier(n_estimators=100)
         ada.fit(x_train, y_train)
```

```
Out[93]: AdaBoostClassifier(n_estimators=100)
```

```
In [94]: print(str(accuracy_score(y_test, ada.predict(x_test))*100)+'%')
100.0%
```

```
In [95]: rf = RandomForestClassifier(n_estimators=100)
         rf.fit(x_train, y_train)
```

```
Out[95]: RandomForestClassifier()
```

```
In [96]: print(str(accuracy_score(y_test, rf.predict(x_test))*100)+'%')
100.0%
```

```
In [ ]:
```