

## Mushroom Classification

```
In [72]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
import joblib

import warnings
warnings.filterwarnings('ignore')

dataset
```

```
In [73]: df = pd.read_csv("mushrooms.csv")
df.shape
```

```
Out[73]: (8124, 23)
```

```
In [74]: df.head()
```

```
Out[74]:
```

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

```
5 rows × 23 columns
```

```
In [75]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8124 entries, 0 to 8123
Data columns (total 23 columns):
 #   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: 1.4+ MB
```

```
In [76]: df.columns
```

```
Out[76]: Index(['class', '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-type', 'veil-color', 'ring-number', 'ring-type', 'spore-print-color', 'population', 'habitat'], dtype='object')
```

```
In [77]: df.isnull().sum()
```

```
Out[77]:
```

|  | class | 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-type | veil-color | ring-number | ring-type | spore-print-color | population | habitat |
|--|-------|-----------|-------------|-----------|---------|------|-----------------|--------------|-----------|------------|-------------|------------|--------------------------|--------------------------|------------------------|------------------------|-----------|------------|-------------|-----------|-------------------|------------|---------|
|  | 0     | 0         | 0           | 0         | 0       | 0    | 0               | 0            | 0         | 0          | 0           | 0          | 0                        | 0                        | 0                      | 0                      | 0         | 0          | 0           | 0         | 0                 | 0          | 0       |

```
In [78]: print(df.value_counts())
```

```
class    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-type    veil-color    ring-number    ring-type    spore-print-color    population    habitat
e        b            f            s            n            t            p            f            c            n            k            ...            2            7            7            0
e        ?            k            g            y            f            n            f            w            n            g            ...            2            7            7            0
p        w            t            p            y            o            l            h            y            d            b            1            w            g            1            w            w            t            p
s        g            1
s        w            n            g            1            w
s        g            1
s        w            t            p            w            w            n            g            1            w            p            w            t            p
s        g            1
s        w            t            p            w            w            n            g            1            w            p            w            n            w            n
p        x            y            b            k            y            f            f            f            c            b            1            7            7            0
e        w            o            l            1            h            y            y            p            1            w            1            7            7            0
g        1
p        1
e        ?            k            e            w            w            n            f            c            w            n            1            7            7            0
y        p            w            o            1            e            w            v            w            v            1            w            1            7            7            0
1        v            w            p            1            o            e            w            v            w            v            1            w            1            7            7            0
Name: count, Length: 8124, dtype: int64
```

```
In [80]: plt.figure(figsize=(5, 4))
sns.countplot(x='class', data=df, palette='Set2')
plt.xlabel('Class (e=edible, p=poisonous)')
plt.show()
```

```
Out[80]: <function matplotlib.pyplot.show(close=None, block=None)>
```

```
In [81]: le= LabelEncoder()
for col in df.columns:
    df[col] = le.fit_transform(df[col])

df.head()
```

```
Out[81]:
```

|   | 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 | ve... |
|---|-------|-----------|-------------|-----------|---------|------|-----------------|--------------|-----------|------------|-----|--------------------------|------------------------|------------------------|-----------|-------|
| 0 | 1     | 5         | 2           | 4         | 1       | 6    | 1               | 0            | 1         | 4          | ... | 2                        | 7                      | 7                      | 0         | v...  |
| 1 | 0     | 5         | 2           | 9         | 1       | 0    | 1               | 0            | 0         | 4          | ... | 2                        | 7                      | 7                      | 0         | v...  |
| 2 | 0     | 0         | 2           | 8         | 1       | 3    | 1               | 0            | 0         | 5          | ... | 2                        | 7                      | 7                      | 0         | v...  |
| 3 | 1     | 5         | 3           | 8         | 1       | 6    | 1               | 0            | 1         | 5          | ... | 2                        | 7                      | 7                      | 0         | v...  |
| 4 | 0     | 5         | 2           | 3         | 0       | 5    | 1               | 1            | 0         | 4          | ... | 2                        | 7                      | 7                      | 0         | v...  |

```
5 rows × 23 columns
```

```
In [82]: le.classes_
```

```
Out[82]: array(['d', 'g', 'l', 'm', 'p', 'u', 'w'], dtype=object)
```

```
In [83]: X = df.drop('class', axis=1)
y = df['class']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print(f"X_train shape : ", X_train.shape)
print(f"X_test shape : ", X_test.shape)
```

```
X_train shape : (6499, 22)
X_test shape : (1625, 22)
```

```
In [84]: rf_model = RandomForestClassifier(n_estimators=200, random_state=42)
rf_model.fit(X_train, y_train)
y_pred = rf_model.predict(X_test)
```

```
Random Forest accuracy: 1.0
precision    recall    f1-score    support
0            1.00    1.00    1.00     843
1            1.00    1.00    1.00     782

accuracy                           1.00     1625
macro avg    1.00    1.00    1.00     1625
weighted avg  1.00    1.00    1.00     1625
```

```
In [85]: confusion_matrix(y_test, y_pred)
```

```
Out[85]: array([[843, 0], [0, 782]])
```

```
In [86]: joblib.dump(rf_model, "random_forest.pkl")
```

```
Out[86]: ['random_forest.pkl']
```