```
import pandas as pd
```

Load data

```
# load the CSV file as a pandas DataFrame
df = pd.read_csv('/content/agaricus-lepiota.data', delimiter=',')
df.columns = ['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']
print(df)
```

```
      class cap-shape cap-surface  ... spore-print-color population habitat
0         e         x           s  ...                 n          n       g
1         e         b           s  ...                 n          n       m
2         p         x           y  ...                 k          s       u
3         e         x           s  ...                 n          a       g
4         e         x           y  ...                 k          n       g
...     ...       ...         ...  ...               ...        ...     ...
8118      e         k           s  ...                 b          c       l
8119      e         x           s  ...                 b          v       l
8120      e         f           s  ...                 b          c       l
8121      p         k           y  ...                 w          v       l
8122      e         x           s  ...                 o          c       l

[8123 rows x 23 columns]
```

Pre-processing

```
# Label encoding
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df_encoded = df.apply(le.fit_transform)
print(df_encoded)
```

```
      class  cap-shape  cap-surface  ...  spore-print-color  population  habitat
0         0          5            2  ...                  3           2        1
1         0          0            2  ...                  3           2        3
2         1          5            3  ...                  2           3        5
3         0          5            2  ...                  3           0        1
4         0          5            3  ...                  2           2        1
...     ...        ...          ...  ...                ...         ...      ...
8118      0          3            2  ...                  0           1        2
8119      0          5            2  ...                  0           4        2
8120      0          2            2  ...                  0           1        2
8121      1          3            3  ...                  7           4        2
8122      0          5            2  ...                  4           1        2

[8123 rows x 23 columns]
```

```
# Extract class column
classes = df_encoded['class']


# Remove class column and 'stalk-root' column (has missing values) from attribute dataframe
df_final = df_encoded.drop(labels=['class', 'stalk-root'], axis=1)


# Split data into train and test data in preparation for classification
from sklearn.model_selection import train_test_split

# where X is the attribute dataframe and Y is the classes dataframe
X_train, X_test, Y_train, Y_test = train_test_split(df_final, classes, test_size=0.2,
                                                    random_state=1)
```

Naive Bayes

```
# Naive Bayes
from sklearn.naive_bayes import CategoricalNB

cNB = CategoricalNB()
cNB.fit(X_train, Y_train)
cNB_predictions = cNB.predict(X_test)
print(cNB_predictions)
#where 0 is edible and 1 is poisonous
```

```
[0 0 1 ... 1 0 0]
```

Support Vector Machine

```
from sklearn import svm

cSVM = svm.SVC()
```

```
cSVM = svm.SVC()
cSVM.fit(X_train, Y_train)
cSVM_predictions = cSVM.predict(X_test)
print(cSVM_predictions)
#where 0 is edible and 1 is poisonous
```

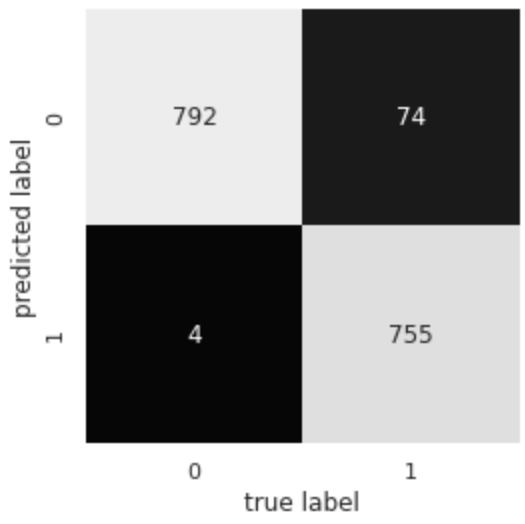
```
[0 1 1 ... 1 0 0]
```

Analysis

```
# Accuracy
from sklearn.metrics import accuracy_score

cNB_accuracy = accuracy_score(Y_test, cNB_predictions)
cSVM_accuracy = accuracy_score(Y_test, cSVM_predictions)

print("Naive Bayes Accuracy:", cNB_accuracy)
print("SVM Accuracy:", cSVM_accuracy)
```

```
Naive Bayes Accuracy: 0.952
SVM Accuracy: 0.9876923076923076
```

```
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()


labels = [0, 1]
mat = confusion_matrix(Y_test, cNB_predictions)
sns.heatmap(mat.T, square=True, annot=True, fmt='d', cbar=False,
            xticklabels=labels, yticklabels=labels)
plt.xlabel('true label')
plt.ylabel('predicted label');
```



```
labels = [0, 1]
mat = confusion_matrix(Y_test, cSVM_predictions)
sns.heatmap(mat.T, square=True, annot=True, fmt='d', cbar=False,
            xticklabels=labels, yticklabels=labels)
plt.xlabel('true label')
plt.ylabel('predicted label');
```