

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

```
df = pd.read_csv('survey lung cancer.csv')
```

```
df.head(3)
```

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING	COUGHING	SHORTNESS OF BREATH
0	M	69	1	2	2	1	1	2	1	2	2	2	2
1	M	74	2	1	1	1	2	2	2	1	1	1	2
2	F	59	1	1	1	2	1	2	1	2	1	2	2

Next steps: [Generate code with df](#) [New interactive sheet](#)

```
df.tail()
```

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING	COUGHING	SHORTNE OF BREA
304	F	56	1	1	1	2	2	2	1	1	2	2	
305	M	70	2	1	1	1	1	2	2	2	2	2	
306	M	58	2	1	1	1	1	1	2	2	2	2	
307	M	67	2	1	2	1	1	2	2	1	2	2	
308	M	62	1	1	1	2	1	2	2	2	2	1	

```
df.sample()
```

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING	COUGHING	SHORTNE OF BREA
297	F	47	2	2	1	2	2	2	2	2	1	2	

```
df.shape
```

(309, 16)

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 309 entries, 0 to 308
Data columns (total 16 columns):
#   Column                                Non-Null Count  Dtype
---  ---                                ---
0   GENDER                                309 non-null    object
1   AGE                                  309 non-null    int64
2   SMOKING                              309 non-null    int64
3   YELLOW_FINGERS                       309 non-null    int64
4   ANXIETY                              309 non-null    int64
5   PEER_PRESSURE                        309 non-null    int64
6   CHRONIC DISEASE                      309 non-null    int64
7   FATIGUE                              309 non-null    int64
8   ALLERGY                              309 non-null    int64
9   WHEEZING                             309 non-null    int64
10  ALCOHOL CONSUMING                    309 non-null    int64
11  COUGHING                             309 non-null    int64
12  SHORTNESS OF BREATH                  309 non-null    int64
13  SWALLOWING DIFFICULTY                309 non-null    int64
14  CHEST PAIN                           309 non-null    int64
15  LUNG_CANCER                          309 non-null    object
dtypes: int64(14), object(2)
memory usage: 38.8+ KB
```



df.describe()

	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING	
count	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000	3
mean	62.673139	1.563107	1.569579	1.498382	1.501618	1.504854	1.673139	1.556634	1.556634	1.556634	
std	8.210301	0.496806	0.495938	0.500808	0.500808	0.500787	0.469827	0.497588	0.497588	0.497588	
min	21.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	
25%	57.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	
50%	62.000000	2.000000	2.000000	1.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	
75%	69.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	
max	87.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	

df.dtypes

	0
GENDER	object
AGE	int64
SMOKING	int64
YELLOW_FINGERS	int64
ANXIETY	int64
PEER_PRESSURE	int64
CHRONIC DISEASE	int64
FATIGUE	int64
ALLERGY	int64
WHEEZING	int64
ALCOHOL CONSUMING	int64
COUGHING	int64
SHORTNESS OF BREATH	int64
SWALLOWING DIFFICULTY	int64
CHEST PAIN	int64
LUNG_CANCER	object

dtype: object

df.isnull().sum()

	0
GENDER	0
AGE	0
SMOKING	0
YELLOW_FINGERS	0
ANXIETY	0
PEER_PRESSURE	0
CHRONIC DISEASE	0
FATIGUE	0
ALLERGY	0
WHEEZING	0
ALCOHOL CONSUMING	0
COUGHING	0
SHORTNESS OF BREATH	0
SWALLOWING DIFFICULTY	0
CHEST PAIN	0
LUNG_CANCER	0

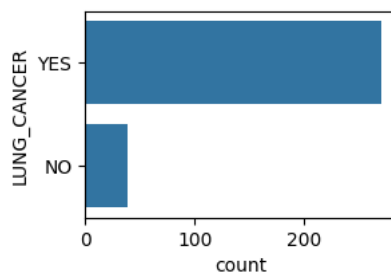
dtype: int64

```
df['LUNG_CANCER'].value_counts()
```

	count
LUNG_CANCER	
YES	270
NO	39

dtype: int64

```
plt.figure(figsize=(3,2))  
sns.countplot(df['LUNG_CANCER'])  
plt.show()
```



✓ encoding

x and y break

train test split

standard scalar

model train

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['LUNG_CANCER'] = le.fit_transform(df['LUNG_CANCER'])
df['GENDER'] = le.fit_transform(df['GENDER'])
```

```
x = df.drop('LUNG_CANCER',axis=1)
y = df['LUNG_CANCER']
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
```

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

```
from sklearn.svm import SVC
model = SVC()
model.fit(x_train,y_train)
```

▼ SVC ⓘ ?

SVC()

```
model.score(x_train,y_train)*100,model.score(x_test,y_test)*100
(94.73684210526315, 96.7741935483871)
```

```
y_pred = model.predict(x_test)
```

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,y_pred)
```

```
cm
array([[ 1,  1],
       [ 1, 59]])
```

```
from sklearn.metrics import accuracy_score, classification_report
print("Accuracy:", accuracy_score(y_test,y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

Accuracy: 0.967741935483871

Classification Report:				
	precision	recall	f1-score	support
0	0.50	0.50	0.50	2
1	0.98	0.98	0.98	60
accuracy			0.97	62
macro avg	0.74	0.74	0.74	62
weighted avg	0.97	0.97	0.97	62

Start coding or [generate](#) with AI.