

```
import numpy as np
import pandas as pd
```

```
df=pd.read_csv('/content/covid_toy - covid_toy (1).csv')
```

```
df.head(2)
```

	age	gender	fever	cough	city	has_covid	
0	60	Male	103.0	Mild	Kolkata	No	
1	27	Male	100.0	Mild	Delhi	Yes	

Next steps:

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

```
df.isnull().sum()
```

	0
age	0
gender	0
fever	10
cough	0
city	0
has_covid	0

dtype: int64

```
df['fever']=df['fever'].fillna(df['fever'].mean())
```

```
df.isnull().sum()
```

	0
age	0
gender	0
fever	0
cough	0
city	0
has_covid	0

dtype: int64

```
from sklearn.preprocessing import LabelEncoder
```

```
lb=LabelEncoder()
```

```
df['city']=lb.fit_transform(df['city'])  
df['cough']=lb.fit_transform(df['cough'])  
df['gender']=lb.fit_transform(df['gender'])
```

```
x=df.drop(columns=['has_covid'])  
y=df['has_covid']
```

```
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.fit_transform(x_test)
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
KKN=KNeighborsClassifier()
```

```
KKN.fit(x_train,y_train)
```

```
▼ KNeighborsClassifier ⓘ ?  
KNeighborsClassifier()
```

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

```
from sklearn.metrics import accuracy_score
```

```
accuracy_score(y_test,y_pred)
```

```
0.45
```

```
df=pd.read_csv('/content/titanic - titanic.csv')
```


```
df.head(2)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cat
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	N
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	N

Next steps:

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

```
df.drop(columns=['Fare','Cabin','Age','Ticket'])
```

	PassengerId	Survived	Pclass	Name	Sex	SibSp	Parch	Embarked	
0	892	0	3	206	1	0	0	Q	
1	893	1	3	403	0	1	0	S	
2	894	0	2	269	1	0	0	Q	
3	895	0	3	408	1	0	0	S	
4	896	1	3	178	0	1	1	S	
...	
413	1305	0	3	353	1	0	0	S	
414	1306	1	1	283	0	0	0	C	
415	1307	0	3	332	1	0	0	S	
416	1308	0	3	384	1	0	0	S	
417	1309	0	3	302	1	1	1	C	

418 rows × 8 columns

```
df.isnull().sum()
```

	0
PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	86
SibSp	0
Parch	0
Ticket	0
Fare	1
Cabin	327
Embarked	0

dtype: int64

```
from sklearn.preprocessing import LabelEncoder
```

```
lb=LabelEncoder()
```

```
df['Name']=lb.fit_transform(df['Name'])
df['Sex']=lb.fit_transform(df['Sex'])
```

```
x=df.drop(columns=['Survived'])
y=df['Survived']
```

```
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.fit_transform(x_test)
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
KKN=KNeighborsClassifier()
```

```
KKN.fit(x_train,y_train)
```

▼ KNeighborsClassifier ⓘ ?
KNeighborsClassifier()

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

```
from sklearn.metrics import accuracy_score
```

```
accuracy_score(y_test,y_pred)
```

```
0.9880952380952381
```

```
df=pd.read_csv('/content/Churn_Modelling - Churn_Modelling.csv')
```

```
df.head(2)
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance
0	1	15634602	Hargrave	619	France	Female	42	2	0.0
1	2	15647311	Hill	608	Spain	Female	41	1	83807.6

Next steps:

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

```
df.isnull().sum()
```

	0
RowNumber	0
CustomerId	0
Surname	0
CreditScore	0
Geography	0
Gender	0
Age	0
Tenure	0
Balance	0
NumOfProducts	0
HasCrCard	0
IsActiveMember	0
EstimatedSalary	0
Exited	0

```
dtype: int64
```

```
from sklearn.preprocessing import LabelEncoder  
lb=LabelEncoder()
```

```
df['Surname']=lb.fit_transform(df['Surname'])  
df['Geography']=lb.fit_transform(df['Geography'])  
df['Gender']=lb.fit_transform(df['Gender'])
```

```
x=df.drop(columns=['Exited'])  
y=df['Exited']
```

```
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.fit_transform(x_test)
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
KKN=KNeighborsClassifier()
```

```
KKN.fit(x_train,y_train)
```

```
▼ KNeighborsClassifier ⓘ ?  
KNeighborsClassifier()
```