


```
import pandas as pd
```

```
df= pd.read_csv('tested.csv')
df
```



	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S
...
413	1305	0	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	S
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C
415	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	S
416	1308	0	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	S
417	1309	0	3	Peter Master Michael J	male	NaN	1	1	2668	22.3583	NaN	C

```
df.describe
```

<bound method NDFrame.describe of				PassengerId	Survived	Pclass	\					
0	892	0	3									
1	893	1	3									
2	894	0	2									
3	895	0	3									
4	896	1	3									
..									
413	1305	0	3									
414	1306	1	1									
415	1307	0	3									
416	1308	0	3									
417	1309	0	3									
				Name	Sex	Age	SibSp	Parch	\			
0				Kelly, Mr. James	male	34.5	0	0				
1				Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0				
2				Myles, Mr. Thomas Francis	male	62.0	0	0				
3				Wirz, Mr. Albert	male	27.0	0	0				
4				Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1				
..							
413				Spector, Mr. Woolf	male	NaN	0	0				
414				Oliva y Ocana, Dona. Fermina	female	39.0	0	0				
415				Saether, Mr. Simon Sivertsen	male	38.5	0	0				
416				Ware, Mr. Frederick	male	NaN	0	0				
417				Peter, Master. Michael J	male	NaN	1	1				
				Ticket	Fare	Cabin	Embarked					
0				330911	7.8292	NaN	Q					
1				363272	7.0000	NaN	S					
2				240276	9.6875	NaN	Q					
3				315154	8.6625	NaN	S					
4				3101298	12.2875	NaN	S					
..								
413				A.5. 3236	8.0500	NaN	S					
414				PC 17758	108.9000	C105	C					
415				SOTON/O.Q. 3101262	7.2500	NaN	S					
416				359309	8.0500	NaN	S					
417				2668	22.3583	NaN	C					

[418 rows x 12 columns]>

```
df.isnull()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	False	False	False	False	False	False	False	False	False	False	True	False
1	False	False	False	False	False	False	False	False	False	False	True	False
2	False	False	False	False	False	False	False	False	False	False	True	False
3	False	False	False	False	False	False	False	False	False	False	True	False
4	False	False	False	False	False	False	False	False	False	False	True	False
...
413	False	False	False	False	False	True	False	False	False	False	True	False
414	False	False	False	False	False	False	False	False	False	False	False	False
415	False	False	False	False	False	False	False	False	False	False	True	False
416	False	False	False	False	False	True	False	False	False	False	True	False
417	False	False	False	False	False	True	False	False	False	False	True	False

418 rows × 12 columns

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

```

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

```

```
df.fillna({'Age': df['Age'].median(), 'Embarked': 'S'}, inplace=True)
```

```

from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='median')
df[['Age', 'Fare']] = imputer.fit_transform(df[['Age', 'Fare']])
df['Embarked'].fillna('S', inplace=True)

```

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

```

label_encoder = LabelEncoder()
df['Sex'] = label_encoder.fit_transform(df['Sex'])
df['Embarked'] = label_encoder.fit_transform(df['Embarked'])

```

```

features = ['Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare', 'Embarked']
X = df[features]
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)

```

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

```

PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age           0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin        327
Embarked       0
dtype: int64

```

```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier(random_state=42)
model.fit(X_train, y_train)
```

```
▼ DecisionTreeClassifier
DecisionTreeClassifier(random_state=42)
```

```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
predictions = model.predict(X_test)
```

```
accuracy = accuracy_score(y_test, predictions)
print(f"Accuracy: {accuracy}")
```

```
Accuracy: 1.0
```

```
print(classification_report(y_test, predictions))
print(confusion_matrix(y_test, predictions))
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	50
1	1.00	1.00	1.00	34
accuracy			1.00	84
macro avg	1.00	1.00	1.00	84
weighted avg	1.00	1.00	1.00	84

```
[[50  0]
 [ 0 34]]
```