In [1]:

```python
import pandas as pd
df=pd.read_csv('train.csv')
df.tail(2)
```

Out[1]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

In [2]:

```python
target=df['Survived']
target
```

Out[2]:

```
0      0
1      1
2      1
3      1
4      0
      ..
886    0
887    1
888    0
889    1
890    0
Name: Survived, Length: 891, dtype: int64
```

In [3]:

```python
df.drop(['Survived'],axis=1,inplace=True)
df.head()
```

Out[3]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

In [4]:

```python
df.drop(['PassengerId','Name','Ticket','Cabin','Embarked'],axis=1,inplace=True)
df.head()
```

Out[4]:

	Pclass	Sex	Age	SibSp	Parch	Fare
0	3	male	22.0	1	0	7.2500
1	1	female	38.0	1	0	71.2833
2	3	female	26.0	0	0	7.9250
3	1	female	35.0	1	0	53.1000
4	3	male	35.0	0	0	8.0500

In [5]:

```python
from sklearn import preprocessing
```

In [6]:

```python
le = preprocessing.LabelEncoder()
mmm=df['Sex']
mmm
```

Out[6]:

```
0        male
1      female
2      female
3      female
4        male
        ...
886      male
887    female
888    female
889      male
890      male
Name: Sex, Length: 891, dtype: object
```

In [7]:

```python
df.drop(['Sex'],axis=1,inplace=True)
df
```

Out[7]:

	Pclass	Age	SibSp	Parch	Fare
0	3	22.0	1	0	7.2500
1	1	38.0	1	0	71.2833
2	3	26.0	0	0	7.9250
3	1	35.0	1	0	53.1000
4	3	35.0	0	0	8.0500
...	...	...	...	...	...
886	2	27.0	0	0	13.0000
887	1	19.0	0	0	30.0000
888	3	NaN	1	2	23.4500
889	1	26.0	0	0	30.0000
890	3	32.0	0	0	7.7500

891 rows × 5 columns

In [8]:

```python
sex = le.fit_transform(mmm)
sex
```

Out[8]:

```
array([1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1,
       0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0,
       0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1,
       0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0,
       1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1,
       0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1,
       1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1,
       1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1,
       0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0,
       1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,
       0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0,
       1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0,
       0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1,
       1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,
       0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1,
       0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0,
       1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1,
       0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0,
       1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1,
       0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1,
       1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1,
       1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0,
       1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0,
       1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0,
       1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1,
       0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1,
       1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1,
       1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1,
       0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0,
       1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1,
       1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1,
       0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0,
       1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0,
       1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1,
       0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0,
       0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1])
```

In [9]:

```python
sex=pd.DataFrame(sex)
sex
```

Out[9]:

	0
0	1
1	0
2	0
3	0
4	1
...	...
886	1
887	0
888	0
889	1
890	1

891 rows × 1 columns

In [10]:

```python
df=pd.concat([sex,df],axis=1)
df
```

Out[10]:

	0	Pclass	Age	SibSp	Parch	Fare
0	1	3	22.0	1	0	7.2500
1	0	1	38.0	1	0	71.2833
2	0	3	26.0	0	0	7.9250
3	0	1	35.0	1	0	53.1000
4	1	3	35.0	0	0	8.0500
...	...	...	...	...	...	...
886	1	2	27.0	0	0	13.0000
887	0	1	19.0	0	0	30.0000
888	0	3	NaN	1	2	23.4500
889	1	1	26.0	0	0	30.0000
890	1	3	32.0	0	0	7.7500

891 rows × 6 columns

In [11]:

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   0       891 non-null    int64  
 1   Pclass  891 non-null    int64  
 2   Age     714 non-null    float64
 3   SibSp   891 non-null    int64  
 4   Parch   891 non-null    int64  
 5   Fare    891 non-null    float64
dtypes: float64(2), int64(4)
memory usage: 41.9 KB
```

In [12]:

```python
df.Age.isna()
```

Out[12]:

```
0      False
1      False
2      False
3      False
4      False
       ...  
886    False
887    False
888     True
889    False
890    False
Name: Age, Length: 891, dtype: bool
```

In [13]:

```python
m=df.Age.mean()
m
```

Out[13]:

```
29.69911764705882
```

In [14]:

```python
df['Age']=df.Age.fillna(m)
```

In [15]:

```python
df
```

Out[15]:

	0	Pclass	Age	SibSp	Parch	Fare
0	1	3	22.000000	1	0	7.2500
1	0	1	38.000000	1	0	71.2833
2	0	3	26.000000	0	0	7.9250
3	0	1	35.000000	1	0	53.1000
4	1	3	35.000000	0	0	8.0500
...	...	...	...	...	...	...
886	1	2	27.000000	0	0	13.0000
887	0	1	19.000000	0	0	30.0000
888	0	3	29.699118	1	2	23.4500
889	1	1	26.000000	0	0	30.0000
890	1	3	32.000000	0	0	7.7500

891 rows × 6 columns

In [16]:

```python
from sklearn.model_selection import train_test_split
```

In [17]:

```python
X_train, X_test, y_train, y_test = train_test_split(df, target, test_size=0.2)
```

In [18]:

```python
len(X_train)
```

Out[18]:

```
712
```

In [19]:

```python
len(X_test)
```

Out[19]:

```
179
```

In [20]:

```python
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=8)
```

In [21]:

```python
knn.fit(X_train, y_train)
```

Out[21]:

```
KNeighborsClassifier(n_neighbors=8)
```

In [22]:

```python
knn.score(X_test, y_test)
```

Out[22]:

```
0.7597765363128491
```

In [23]:

```python
knn.predict([[1,1,24,0,0,30]])
```

Out[23]:

```
array([1])
```