```
import pandas as pd
import numpy as np
import random
from sklearn import preprocessing
le = preprocessing.LabelEncoder()


from sklearn.neighbors import KNeighborsClassifier


dataFile = pd.read_csv('Titanic_train.csv')


dataFile.head()
```

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 3 | Braund, Mr. Owen Harris | male | 22.0 | 1 | 0 | A/5 21171 | |
| **1** | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence | female | 38.0 | 1 | 0 | PC 17599 | 7 |

```
dataFile = dataFile[["Survived","Pclass","Age", "Sex","SibSp","Parch","Fare","Embarked"]]


dataFile=dataFile.dropna()


dataFile.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 712 entries, 0 to 890
Data columns (total 8 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Survived  712 non-null    int64
 1   Pclass    712 non-null    int64
 2   Age       712 non-null    float64
 3   Sex       712 non-null    object
 4   SibSp     712 non-null    int64
 5   Parch     712 non-null    int64
 6   Fare      712 non-null    float64
 7   Embarked  712 non-null    object
dtypes: float64(2), int64(4), object(2)
memory usage: 50.1+ KB
```

```
pclass  =  list(dataFile["Pclass"])
sex  =  list(dataFile["Sex"])
age =  list(dataFile["Age"])
sibSp =   list(dataFile["SibSp"])
parch =  list(dataFile["Parch"])
fare = list(dataFile["Fare"])
#cabin =   list(dataFile["Cabin"])
embarked =   list(dataFile["Embarked"])

survived=list(dataFile["Survived"])


pclass_encoded=le.fit_transform(pclass)
sex_encoded=le.fit_transform(sex)
age_encoded=le.fit_transform(age)
sibSp_encoded=le.fit_transform(sibSp)
parch_encoded=le.fit_transform(parch)
fare_encoded=le.fit_transform(fare)
#cabin_encoded=le.fit_transform(cabin)
embarked_encoded=le.fit_transform(embarked)
survived_encoded=le.fit_transform(survived)



data=list(zip(pclass_encoded,sex_encoded,age_encoded,sibSp_encoded,parch_encoded,embarked_encoded,fare_encoded))#,cabin_encoded))
print(data)
```

```
[(2, 1, 28, 1, 0, 2, 16), (0, 0, 51, 1, 0, 0, 180), (2, 0, 34, 0, 0, 2, 32), (0, 0, 47, 1, 0, 2, 163), (2, 1, 47, 0, 0, 2, 34), (0,
```

```
label=le.fit_transform(survived_encoded)
print(label)
```

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

```
from sklearn.model_selection import train_test_split


x_train, x_test, y_train, y_test = train_test_split(data, label, test_size=570)


for i in range (3,38):
  neigh = KNeighborsClassifier(n_neighbors=i)
  neigh.fit(x_train,y_train)
  neigh.score(x_test,y_test)
  print(i, " " ,neigh.score(x_test,y_test))
```

```
3   0.6614035087719298
4   0.656140350877193
5   0.6543859649122807
6   0.6736842105263158
7   0.6684210526315789
8   0.6578947368421053
9   0.6614035087719298
10   0.6719298245614035
11   0.6859649122807018
12   0.6684210526315789
13   0.6719298245614035
14   0.656140350877193
15   0.6701754385964912
16   0.6701754385964912
17   0.6736842105263158
18   0.6789473684210526
19   0.6684210526315789
20   0.6754385964912281
21   0.6754385964912281
22   0.6736842105263158
23   0.6736842105263158
24   0.6859649122807018
25   0.6807017543859649
26   0.6719298245614035
27   0.6701754385964912
28   0.6736842105263158
29   0.6701754385964912
30   0.6666666666666666
31   0.6631578947368421
32   0.6649122807017543
33   0.6649122807017543
34   0.6526315789473685
35   0.6543859649122807
36   0.6491228070175439
37   0.6473684210526316
```

```
0.7901591895803184
```

✓ 2 s ukończono o 15:50