

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split, KFold
from sklearn.model_selection import KFold, StratifiedKFold, cross_val_score
import matplotlib.pyplot as plt
```

```
In [2]: df=pd.read_csv('titanica.csv')
df
```

Out[2]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	NaN	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN
1	893.0	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN
2	894.0	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN
3	895.0	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN
4	896.0	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN
...
413	1305.0	0	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN
414	1306.0	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C1
415	1307.0	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN
416	1308.0	0	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN
417	1309.0	0	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN

418 rows × 12 columns



In [3]: df.head()

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	NaN	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	
1	893.0	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	
2	894.0	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	
3	895.0	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	
4	896.0	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	

In [4]:

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
Column Non-Null Count Dtype
--- -
0 PassengerId 417 non-null float64
1 Survived 418 non-null int64
2 Pclass 418 non-null int64
3 Name 418 non-null object
4 Sex 418 non-null object
5 Age 332 non-null float64
6 SibSp 418 non-null int64
7 Parch 418 non-null int64
8 Ticket 418 non-null object
9 Fare 417 non-null float64
10 Cabin 91 non-null object
11 Embarked 418 non-null object
dtypes: float64(3), int64(4), object(5)
memory usage: 39.3+ KB

In [5]:

df.isnull().sum()

```
Out[5]: PassengerId      1  
Survived      0  
Pclass        0  
Name          0  
Sex           0  
Age           86  
SibSp         0  
Parch         0  
Ticket        0  
Fare          1  
Cabin        327  
Embarked      0  
dtype: int64
```

```
In [6]: df.dropna(axis=0, how='any', subset=None, inplace=False)
```

Out[6]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cab
12	904.0	1	1	Snyder, Mrs. John Pillsbury (Nelle Stevenson)	female	23.0	1	0	21228	82.2667	B4
14	906.0	1	1	Chaffee, Mrs. Herbert Fuller (Carrie Constance...	female	47.0	1	0	W.E.P. 5734	61.1750	E3
24	916.0	1	1	Ryerson, Mrs. Arthur Larned (Emily Maria Borie)	female	48.0	1	3	PC 17608	262.3750	B1 B1 B6 B6
26	918.0	1	1	Ostby, Miss. Helene Ragnhild	female	22.0	0	1	113509	61.9792	B3
28	920.0	0	1	Brady, Mr. John Bertram	male	41.0	0	0	113054	30.5000	A2
...
404	1296.0	0	1	Frauenthal, Mr. Isaac Gerald	male	43.0	1	0	17765	27.7208	D4
405	1297.0	0	2	Nourney, Mr. Alfred (Baron von Drachstedt)"	male	20.0	0	0	SC/PARIS 2166	13.8625	D3
407	1299.0	0	1	Widener, Mr. George Dunton	male	50.0	1	1	113503	211.5000	C8
411	1303.0	1	1	Minahan, Mrs. William Edward (Lillian E Thorpe)	female	37.0	1	0	19928	90.0000	C1
414	1306.0	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C10

87 rows × 12 columns

◀

▶

```
In [7]: y=df['Survived']
x=df.drop(['Survived'],axis=1)
y
```

```
Out[7]: 0      0
        1      1
        2      0
        3      0
        4      1
        ..
        413    0
        414    1
        415    0
        416    0
        417    0
        Name: Survived, Length: 418, dtype: int64
```

```
In [8]: x.head()
```

Out[8]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	NaN	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893.0	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	896.0	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S



```
In [9]: df=df.drop(['Name','Sex','Embarked','Ticket','Cabin'],axis=1)
```

```
In [10]: df
```

Out[10]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
0	NaN	0	3	34.5	0	0	7.8292
1	893.0	1	3	47.0	1	0	7.0000
2	894.0	0	2	62.0	0	0	9.6875
3	895.0	0	3	27.0	0	0	8.6625
4	896.0	1	3	22.0	1	1	12.2875
...
413	1305.0	0	3	NaN	0	0	8.0500
414	1306.0	1	1	39.0	0	0	108.9000
415	1307.0	0	3	38.5	0	0	7.2500
416	1308.0	0	3	NaN	0	0	8.0500
417	1309.0	0	3	NaN	1	1	22.3583

418 rows × 7 columns

In [11]:

```
kf=KFold(n_splits=5)
print("Data is split into following number of folds:")
kf.get_n_splits(df)
```

Data is split into following number of folds:

Out[11]:

5

In [12]:

```
clf=DecisionTreeClassifier()
print("Accuracies for each fold of data are:")
for train_index, test_index in kf.split(df,y):
    clf.fit(df.iloc[train_index], y.iloc[train_index])
    pred=clf.predict(df.iloc[test_index,:])
    print(round(accuracy_score(y.iloc[test_index],pred),3))
```

Accuracies for each fold of data are:

```
1.0
1.0
1.0
1.0
1.0
```

In [13]:

```
print(df)
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
0	NaN	0	3	34.5	0	0	7.8292
1	893.0	1	3	47.0	1	0	7.0000
2	894.0	0	2	62.0	0	0	9.6875
3	895.0	0	3	27.0	0	0	8.6625
4	896.0	1	3	22.0	1	1	12.2875
..
413	1305.0	0	3	NaN	0	0	8.0500
414	1306.0	1	1	39.0	0	0	108.9000
415	1307.0	0	3	38.5	0	0	7.2500
416	1308.0	0	3	NaN	0	0	8.0500
417	1309.0	0	3	NaN	1	1	22.3583

[418 rows x 7 columns]

```
In [14]: cv =cross_val_score(DecisionTreeClassifier(),df,y,cv=5,scoring='accuracy')
print("Accuracies of all the folds after the cross validation are:")
cv
```

Accuracies of all the folds after the cross validation are:

```
Out[14]: array([1., 1., 1., 1., 1.])
```

In []: