In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
```

In [2]:
```
df = pd.read_csv('train.csv')[['Age','Pclass','SibSp','Parch','Survived']]
df.head()
```

Out[2]:

| | Age | Pclass | SibSp | Parch | Survived |
|---|---|---|---|---|---|
| **0** | 22.0 | 3 | 1 | 0 | 0 |
| **1** | 38.0 | 1 | 1 | 0 | 1 |
| **2** | 26.0 | 3 | 0 | 0 | 1 |
| **3** | 35.0 | 1 | 1 | 0 | 1 |
| **4** | 35.0 | 3 | 0 | 0 | 0 |

In [3]:
```
df.dropna(inplace=True)
df.head()
```

Out[3]:

| | Age | Pclass | SibSp | Parch | Survived |
|---|---|---|---|---|---|
| **0** | 22.0 | 3 | 1 | 0 | 0 |
| **1** | 38.0 | 1 | 1 | 0 | 1 |
| **2** | 26.0 | 3 | 0 | 0 | 1 |
| **3** | 35.0 | 1 | 1 | 0 | 1 |
| **4** | 35.0 | 3 | 0 | 0 | 0 |

In [4]:
```
x = df.iloc[:,0:4]
y = df.iloc[:,-1]
```

In [5]:
```
x.head()
```

Out[5]:

| | Age | Pclass | SibSp | Parch |
|---|---|---|---|---|
| **0** | 22.0 | 3 | 1 | 0 |
| **1** | 38.0 | 1 | 1 | 0 |
| **2** | 26.0 | 3 | 0 | 0 |
| **3** | 35.0 | 1 | 1 | 0 |
| **4** | 35.0 | 3 | 0 | 0 |

In [6]:
```
y.head()
```

Out[6]:
```
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

In [7]:
```
np.mean(cross_val_score(LogisticRegression(),x,y,scoring='accuracy',cv=20))
```

Out[7]:
```
0.6933333333333332
```

In [8]:
```
cross_val_score(LogisticRegression(),x,y,scoring='accuracy',cv=20)
```

Out[8]:
```
array([0.61111111, 0.63888889, 0.61111111, 0.55555556, 0.77777778,
       0.55555556, 0.80555556, 0.63888889, 0.72222222, 0.72222222,
       0.72222222, 0.72222222, 0.75      , 0.83333333, 0.54285714,
       0.88571429, 0.68571429, 0.68571429, 0.74285714, 0.65714286])
```

In [9]:
```
x['Family_size'] = x['SibSp'] + x['Parch'] + 1
x.head()
```

Out[9]:

| | Age | Pclass | SibSp | Parch | Family_size |
|---|---|---|---|---|---|
| **0** | 22.0 | 3 | 1 | 0 | 2 |
| **1** | 38.0 | 1 | 1 | 0 | 2 |
| **2** | 26.0 | 3 | 0 | 0 | 1 |
| **3** | 35.0 | 1 | 1 | 0 | 2 |
| **4** | 35.0 | 3 | 0 | 0 | 1 |

In [11]:
```
def myfunc(num):
    if num == 1:
     # alone
        return 0
    elif num >1 and num <=4:
     #small family
        return 1
    else:
     # large family
        return 2

myfunc(4)
```

Out[11]:
```
1
```

In [12]:
```
x['Famuly_type'] = x['Family_size'].apply(myfunc)
```

In [13]:
```
x.head()
```

Out[13]:

| | Age | Pclass | SibSp | Parch | Family_size | Famuly_type |
|---|---|---|---|---|---|---|
| **0** | 22.0 | 3 | 1 | 0 | 2 | 1 |
| **1** | 38.0 | 1 | 1 | 0 | 2 | 1 |
| **2** | 26.0 | 3 | 0 | 0 | 1 | 0 |
| **3** | 35.0 | 1 | 1 | 0 | 2 | 1 |
| **4** | 35.0 | 3 | 0 | 0 | 1 | 0 |

In [14]:
```
x.drop(columns=['SibSp','Parch','Family_size'],inplace=True)
```

In [15]:
```
x.head()
```

Out[15]:

| | Age | Pclass | Famuly_type |
|---|---|---|---|
| **0** | 22.0 | 3 | 1 |
| **1** | 38.0 | 1 | 1 |
| **2** | 26.0 | 3 | 0 |
| **3** | 35.0 | 1 | 1 |
| **4** | 35.0 | 3 | 0 |

In [16]:
```
np.mean(cross_val_score(LogisticRegression(),x,y,scoring='accuracy',cv=20))
```

Out[16]:
```
0.7003174603174602
```

# Feature Splitting

In [17]:
```
df = pd.read_csv('train.csv')
df.head()
```

Out[17]:

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

In [18]:
```
df['Name']
```

Out[18]:
```
0                                Braund, Mr. Owen Harris
1      Cumings, Mrs. John Bradley (Florence Briggs Th...
2                                 Heikkinen, Miss. Laina
3           Futrelle, Mrs. Jacques Heath (Lily May Peel)
4                               Allen, Mr. William Henry
                             ...
886                                Montvila, Rev. Juozas
887                         Graham, Miss. Margaret Edith
888             Johnston, Miss. Catherine Helen "Carrie"
889                                Behr, Mr. Karl Howell
890                                  Dooley, Mr. Patrick
Name: Name, Length: 891, dtype: object
```

In [19]:
```
df['Title'] = df['Name'].str.split(',', expand=True)[1].str.split('.', expand=

df[['Title','Name']]
```

Out[19]:

| | Title | Name |
|---|---|---|
| **0** | Mr | Braund, Mr. Owen Harris |
| **1** | Mrs | Cumings, Mrs. John Bradley (Florence Briggs Th... |
| **2** | Miss | Heikkinen, Miss. Laina |
| **3** | Mrs | Futrelle, Mrs. Jacques Heath (Lily May Peel) |
| **4** | Mr | Allen, Mr. William Henry |
| **...** | ... | ... |
| **886** | Rev | Montvila, Rev. Juozas |
| **887** | Miss | Graham, Miss. Margaret Edith |
| **888** | Miss | Johnston, Miss. Catherine Helen "Carrie" |
| **889** | Mr | Behr, Mr. Karl Howell |
| **890** | Mr | Dooley, Mr. Patrick |

891 rows × 2 columns

In [20]:
```
(df.groupby('Title').mean()['Survived']).sort_values(ascending=False)
```

Out[20]:
```
Title
 the Countess    1.000000
 Mlle            1.000000
 Sir             1.000000
 Ms              1.000000
 Lady            1.000000
 Mme             1.000000
 Mrs             0.792000
 Miss            0.697802
 Master          0.575000
 Col             0.500000
 Major           0.500000
 Dr              0.428571
 Mr              0.156673
 Jonkheer        0.000000
 Rev             0.000000
 Don             0.000000
 Capt            0.000000
Name: Survived, dtype: float64
```

In [21]:
```
df['Is_Married'] = 0
df['Is_Married'].loc[df['Title']=='Mrs'] = 1
```

```
C:\Users\rawat\AppData\Local\Temp\ipykernel_50880\3467334201.py:2: SettingWit
hCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  df['Is_Married'].loc[df['Title']=='Mrs'] = 1
```

In [22]:
```
df['Is_Married']
```

Out[22]:
```
0      0
1      0
2      0
3      0
4      0
      ..
886    0
887    0
888    0
889    0
890    0
Name: Is_Married, Length: 891, dtype: int64
```

## Conclusion: From feature construction by using combing and spliting features survival status of each categories is given above.

In [ ]:
```

```