```
Aim: Demonstrate the working of feature construction by combining and spliting the features to extraction the information from
the dataset and write a conclusion about survivals status of different age group
```

In [8]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
```

In [9]:
```
df=pd.read_csv('DATA/train.csv')[['Age','Pclass','SibSp','Parch','Survived']]
df.head()
```

Out[9]:

| | 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 [10]:
```
df.dropna(inplace=True)
```

In [11]:
```
df.head()
```

Out[11]:

| | 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 [12]:
```
X = df.iloc[:,0:4]
y = df.iloc[:,-1]
```

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

Out[13]:

| | 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 [15]:
```
np.mean(cross_val_score(LogisticRegression(),X,y,scoring="accuracy",cv=20))
```

Out[15]:
```
0.6933333333333332
```

In [17]:
```
X['Family_size'] = X['SibSp']+X['Parch']+1
```

In [18]:
```
X.head()
```

Out[18]:

| | 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 [21]:
```
cross_val_score(LogisticRegression(),X,y,scoring="accuracy",cv=20)
```

Out[21]:
```
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 [29]:
```
def myfunc(num):
    if num == 1:
        return 0
    elif num > 1 and num <=4:
        return 1
    else:
        return 2
```

In [30]:
```
myfunc(4)
```

Out[30]:
```
1
```

In [31]:
```
X['Family_type']= X['Family_size'].apply(myfunc)
```

In [32]:
```
X.head()
```

Out[32]:

| | Age | Pclass | SibSp | Parch | Family_size | Family_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 [33]:
```
np.mean(cross_val_score(LogisticRegression(),X,y,scoring="accuracy",cv=20))
```

Out[33]:
```
0.7031746031746031
```

In [34]:
```
df=pd.read_csv('DATA/train.csv')
```

In [35]:
```
df.head()
```

Out[35]:

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

In [42]:
```
df['Title']=df['Name'].str.split(', ',expand=True)[1].str.split('.',expand=True)[0]
```

In [45]:
```
df[['Title','Name']]
```

Out[45]:

| | 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 [48]:
```
(df.groupby('Title').mean()['Survived']).sort_values(False)
```

```
C:\Users\User23\AppData\Local\Temp\ipykernel_12820\2479167924.py:1: FutureWarning: In a future version of pandas all arguments
of Series.sort_values will be keyword-only.
  (df.groupby('Title').mean()['Survived']).sort_values(False)
```

Out[48]:
```
Title
Capt            0.000000
Don             0.000000
Jonkheer        0.000000
Rev             0.000000
Mr              0.156673
Dr              0.428571
Col             0.500000
Major           0.500000
Master          0.575000
Miss            0.697802
Mrs             0.792000
Mme             1.000000
Sir             1.000000
Ms              1.000000
Lady            1.000000
Mlle            1.000000
the Countess    1.000000
Name: Survived, dtype: float64
```

In [49]:
```
df['Is_married']=0
df['Is_married'].loc[df["Title"]=='Mrs']=1
```

```
C:\Users\User23\AppData\Local\Temp\ipykernel_12820\2148996725.py:2: SettingWithCopyWarning:
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/stable/user_guide/indexing.html#returning-a-view-ve
rsus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  df['Is_married'].loc[df["Title"]=='Mrs']=1
```

In [50]:
```
df['Is_married']
```

Out[50]:
```
0      0
1      1
2      0
3      1
4      0
      ..
886    0
887    0
888    0
889    0
890    0
Name: Is_married, Length: 891, dtype: int64
```

Conclusion: From the above expriment we conclude that the death rate of higher class people was nearly zero and deaths of nobel males was highest they secrificed themselves to save others the rate of child and ladies was also low.

In [ ]: