Aim:Demonstrate the working of feature construction by combining at splitting the features to extract the information from the dataset and wite a conclusion about survival status of different salutation. ¶

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
```

In [2]:

```
df = pd.read_csv('train - train (1).csv')[['Age','Pclass','SibSp','Parch','Survived']]
```

In [3]:

```
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]:

```
df.dropna(inplace=True)
```

In [5]:

```
df.head()
```

Out[5]:

	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 [6]:

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

In [7]:

```
x.head()
```

Out[7]:

	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 [13]:

```
(cross_val_score(LogisticRegression(),x,y,scoring='accuracy',cv=20))def myfunc()
```

Out[13]:

```
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 [14]:

```
np.mean(cross_val_score(LogisticRegression(),x,y,scoring='accuracy',cv=20))
```

Out[14]:

```
0.6933333333333332
```

Applying Feature construction

In [10]:

```
x['Family_size'] = x['SibSp'] + x['Parch'] +1
```

In [11]:

```
x.head()
```

Out[11]:

	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 [17]:

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

In [19]:

```
myfunc(4)
```

Out[19]:

```
1
```

In [20]:

```
x['Family_type'] = x['Family_size'].apply(myfunc)
```

In [21]:

```
x.head()
```

Out[21]:

	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 [24]:

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

In [25]:

```
x.head()
```

Out[25]:

	Age	Pclass	Family_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 [26]:

```
np.mean(cross_val_score(LogisticRegression(),x,y,scoring='accuracy',cv=20))
```

Out[26]:

```
0.7003174603174602
```

Feature Splitting

In [27]:

```
df = pd.read_csv('train - train (1).csv')
```

In [28]:

```
df.head()
```

Out[28]:

	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 [29]:

```
df['Name']
```

Out[29]:

```
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 [30]:

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

In [37]:

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

Out[37]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Title
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S	Mr
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C	Mrs
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	Miss
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S	Mrs
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S	Mr
...	...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S	Rev
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S	Miss
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S	Miss
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C	Mr
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q	Mr

891 rows × 13 columns

In [38]:

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

Out[38]:

	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 [49]:

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

```
C:\Users\User14\AppData\Local\Temp\ipykernel_11652\2479167924.py:1: FutureWarning: In a future version of pandas all argume
nts of Series.sort_values will be keyword-only.
  (df.groupby('Title').mean()['Survived']).sort_values(False)
```

Out[49]:

```
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 [50]:

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

```
C:\Users\User14\AppData\Local\Temp\ipykernel_11652\2254989826.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-vie
w-versus-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 [51]:

```
df['Is_Married']
```

Out[51]:

```
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 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 [ ]: