

# Read titanic dataset, predict for following test cases whether person will survive or not. Assume necessary features if necessary.

1) Class : 1, Gender : Female, Age : 42, Siblings / Spouses : 2, Parents/Children: 1 2)  
Class : 3, Gender : male, Age : 30, Siblings / Spouses : 0, Parents/Children: 0

```
In [ ]: import pandas as pd
```

```
In [ ]: df_train = pd.read_csv("train.csv")  
df_train.head()
```

```
In [ ]: df_train.shape
```

```
In [ ]: df_test = pd.read_csv("test.csv")  
df_test.head()
```

```
In [ ]: df_test.shape
```

```
In [ ]: df = df_train  
df.info()
```

```
In [ ]: df.isnull().sum()
```

```
In [ ]: df.drop(['Cabin'], inplace = True, axis = 1)
```

```
In [ ]: df['Age'].fillna(df['Age'].mean(), inplace = True)
```

```
In [ ]: df['Embarked'].unique()
```

```
In [ ]: df['Embarked'].fillna(df['Embarked'].mode()[0], inplace = True)
```

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

```
In [ ]: df.drop(["Name", "PassengerId", "Ticket", "Fare"], axis = 1, inplace= True)
```

```
In [ ]: from sklearn.preprocessing import LabelEncoder  
  
le = LabelEncoder()  
cat_list = df.select_dtypes(include=object).columns.to_list()  
for col in cat_list:  
    df[col] = le.fit_transform(df[col])
```

```
In [ ]: Y = df['Survived']  
X = df.drop('Survived', axis = 1)
```

```
In [ ]: from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X,Y,test_size = 0.3,
```

```
In [ ]: from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(max_iter = 1000)
lr.fit(x_train, y_train)
```

```
In [ ]: y_pred = lr.predict(x_test)
```

```
In [ ]: from sklearn.metrics import confusion_matrix,classification_report
print(classification_report(y_test, y_pred))
```

```
In [ ]: confusion_matrix(y_test, y_pred)
```

```
In [ ]: X.columns
```

```
In [ ]: X.Sex.unique()
```

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

```
In [ ]: #Class : 1, Gender : Female, Age : 42, Siblings / Spouses : 2, Parents/Ch
#Class : 3, Gender : male, Age : 30, Siblings / Spouses : 0, Parents/Chil
data = [[1,0,42,2,1,2], [3,1,30,0,0,2]]
test = pd.DataFrame(data = data, columns = ['Pclass','Sex','Age','SibSp',
```

```
In [ ]: y_pred = lr.predict(test)
y_pred
```