Code Implementation:

Answer:

1) Libraries:

```
2) import numpy as np
3) import pandas as pd
4) import matplotlib.pyplot as plt
5) import seaborn as sns
6) from sklearn.preprocessing import LabelEncoder
7) from sklearn.model_selection import train_test_split
8) from xgboost import XGBRegressor
9) from sklearn import metrics
```

2)Data Collection and processing:

```
Titanic_data = pd.read_csv("Train.csv")
print(Titanic_data)
```

```
Titanic_data.head(2)
```

```
Titanic_data.tail(2)
```

```
Titanic_data.shape
```

```
Titanic_data.info()
```

3) Catagorical Featureset_Type:

```
Titanic_data.isnull().sum()
```

```
Titanic_data['Age'].mean()
```

```
Titanic_data['Age'].fillna(Titanic_data['Age'].mean(), inplace=True)
```

```
Titanic_data['Cabin'].mode()
```

```
mode_of_Cabin = mode_of_Cabin =Titanic_data .pivot_table(values='Cabin', columns='Sex', aggfunc=(lambda x: x.mode()[0]))
print(mode_of_Cabin)
```

```
miss_values =Titanic_data ['Cabin'].isnull()
```

```
print(miss_values)
```

```
Titanic_data.loc[miss_values, 'Cabin'] = Titanic_data.loc[miss_values,'Sex'].apply(lambda x: mode_of_Cabin[x])
```

```
Titanic_data.isnull().sum()
```

```
Titanic_data['Embarked'].mode()
```

```
mode_of_embarked = mode_of_embarked =Titanic_data .pivot_table(values='Embarked', columns='Survived', aggfunc=(lambda x: x.mode()[0]))
```

```
print(mode_of_embarked)
```

```
miss_values =Titanic_data ['Embarked'].isnull()
```

```
 print(miss_values)
```

```
Titanic_data.loc[miss_values, 'Embarked'] = Titanic_data.loc[miss_values,'Sex'].apply(lambda x: mode_of_Cabin[x])
```

```
Titanic_data.isnull().sum()
```

4) **Data Analysis:**

```
Titanic_data.describe()
```

```
sns.set()
```

```
plt.figure(figsize=(6,6))
```

```
sns.histplot(Titanic_data['Age'])
plt.show()
```

```
plt.figure(figsize=(6,6))
sns.histplot(Titanic_data['Sex'])
plt.show()
```

```
plt.figure(figsize=(6,6))
sns.histplot(Titanic_data['Fare'])
plt.show()
```

```
plt.figure(figsize=(6,6))
sns.countplot(x='Survived', data=Titanic_data)
plt.show()
```

```
plt.figure(figsize=(6,6))
sns.countplot(x='Pclass', data=Titanic_data)
plt.show()
```

5) Data pre-processing:

```
Titanic_data.head()
```

```
Titanic_data['Sex'].value_counts()
```

```
Titanic_data.replace({'Sex': {'male':'Male', 'female':'Female'}}, inplace=True)
```

```
Titanic_data['Sex'].value_counts()
```

6) Label-Encoding:

```
encoder = LabelEncoder()
```

```
Titanic_data['Name'] = encoder.fit_transform(Titanic_data['Name'])
Titanic_data['Sex'] = encoder.fit_transform(Titanic_data['Sex'])
Titanic_data['Ticket'] = encoder.fit_transform(Titanic_data['Ticket'])
Titanic_data['Cabin'] = encoder.fit_transform(Titanic_data['Cabin'])
Titanic_data['Embarked'] = encoder.fit_transform(Titanic_data['Embarked'])
```

```
Titanic_data.head()
```

7) Spliting Features and Target:

```
X = Titanic_data.drop(columns='Cabin', axis=1)
Y = Titanic_data['Cabin']
```

```
print(X)
```

```
print(Y)
```

8) Splitting data into training data & Testing data:

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,
random_state=112)
```

```
print("Training data shape:", X_train.shape, Y_train.shape)
print("Testing data shape:", X_test.shape, Y_test.shape)
```

10) Machine learning model training:

```
regressor = XGBRegressor()
```

```
   regressor.fit(X_train, Y_train)
```

11) Evaluation:

```
training_data_prediction = regressor.predict(X_train)
```

```
r2_train = metrics.r2_score(Y_train, training_data_prediction)
```

```
print('R Squared value = ', r2_train)
```

```
test_data_prediction = regressor.predict(X_test)
```

```
r2_test = metrics.r2_score(Y_test, test_data_prediction)
```

Lab Task :