# Green University of Bangladesh

## Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering
Semester: (Summer, Year: 2021), B.Sc. in CSE (Day)

LAB REPORT NO 1
Course Title: Data Mining Lab
Course Code: CSE 424 Section: 191 D1

**Lab Experiment Name:** Handle the null value (missing values) from the attached dataset

Student Details

| Name | | ID |
|---|---|---|
| 1. | Mohammad Rifat Noor | 191002144 |

Lab Date : 03-11-2022
Submission Date : 10-11-2022
Course Teacher's Name : Sadia Afroze

[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status
Marks: ..... Signature: .....
Comments: ..... Date: .....

## 1. TITLE OF THE LAB EXPERIMENT

Handle the null value (missing values) from the attached dataset.

## 2. OBJECTIVES

We've been given a dataset called train (2).CSV. Our goal is to handle the null & empty values.

The dataset contains total 12 columns with multiple columns containing null or empty values.

To handle this issue we are going to use python programing language in google colab. We are also going to use some famous python libraries like pandas & numpy.

## 3. IMPLEMENTATION IN Python:

```
import pandas as pd
import numpy as np
main_df = pd.read_csv("train.csv")
df = main_df
# print(df.head())

df.info()

df['Age'] = df['Age'].fillna(df['Age'].mean())
df['Cabin'] = df['Cabin'].fillna("X")
df['Embarked'] = df['Embarked'].fillna("X")

df.info()

print(df.isnull().sum())
```

## 4. TEST RESULT / OUTPUT:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          891 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        891 non-null    object
 11  Embarked     891 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
```

```
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

## 5. ANALYSIS AND DISCUSSION

First we have to identify the columns that contains null values. To do that we need to use `df.info() function`.

The dataset has 3 columns that contains null values the Age, Cabin & Embarked. We can fill the Age with the mean value of other age values. But for Cabin & Embarked we need to use some other method since they contain character string as their value.

For Cabin & Embarked the best option is to fill the null values with a impossible or unusual value so that we can identify it later. So we replaced all the null value with character “X”.