



**Daffodil**  
*International*  
**University**

## **Lab Report**

**Course Code:** CSE 316

**Course Name:** Artificial Intelligence Lab

**Lab Report No:** 03

**Date of submission :** 15-08-2023

**Submitted To :**

Md. Sadekur Rahman

Assistant Professor

Department of CSE

Daffodil International University

**Submitted By:**

**Name:** Rayhan Rafin

**ID:** 213-15-4278

**Section:** 60\_B

<b>No</b>	<b>Content</b>	<b>Page No</b>
1.	Getting started	3
2.	Checking Null Values	3
3.	Null Value Handling	4
4.	Label Encoder	6
5.	OneHotEncoder	7

## Getting started

### Checking original data:

```
#checking the original data
file_path = "/content/train.csv"

import numpy as np
import pandas as pd

df = pd.read_csv(file_path)
df
```

	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
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

## Checking Null Values

### Check null:

```
# Here i checked if there is any null value present in the data
df.isnull().sum()
```

```
PassengerId      0
Survived          0
Pclass           0
Name             0
Sex              0
Age             177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin           687
Embarked         2
dtype: int64
```

## Check null percentage:

```
'''By dividing the total null values  
of the data by the total length  
we can get the percentage of null'''
```

```
(df.isnull().sum())/len(df)*100
```

```
PassengerId    0.000000  
Survived        0.000000  
Pclass          0.000000  
Name            0.000000  
Sex             0.000000  
Age            19.865320  
SibSp           0.000000  
Parch           0.000000  
Ticket          0.000000  
Fare            0.000000  
Cabin          77.104377  
Embarked        0.224467  
dtype: float64
```

## Null Value Handling

### Dropna Function:

```
#using dropna function removes all the null values  
df_1 = df.copy()  
df_1 = df_1.dropna()  
df_1
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	PP 9549	16.7000	G6	S
11	12	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783	26.5500	C103	S
...	...	...	...	...	...	...	...	...	...	...	...	...
871	872	1	1	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47.0	1	1	11751	52.5542	D35	S
872	873	0	1	Carlsson, Mr. Frans Olof	male	33.0	0	0	695	5.0000	B51 B53 B55	S
879	880	1	1	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56.0	0	1	11767	83.1583	C50	C
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C

183 rows x 12 columns

```
#null percentage after dropna function
(df_1.isnull().sum())/len(df_1)*100
```

PassengerId	0.0
Survived	0.0
Pclass	0.0
Name	0.0
Sex	0.0
Age	0.0
SibSp	0.0
Parch	0.0
Ticket	0.0
Fare	0.0
Cabin	0.0
Embarked	0.0

dtype: float64

### **Fillna Function:**

```
#Replacing the null values of age column with 1001
df_2 = df.copy()
df_2['Age'].fillna(1001,inplace = True)
df_2['Age']
```

0	22.0
1	38.0
2	26.0
3	35.0
4	35.0
...	
886	27.0
887	19.0
888	1001.0
889	26.0
890	32.0

Name: Age, Length: 891, dtype: float64

```
# Replaced the null values with the mean of age column
df_x = df.copy()
df_x['Age'].fillna(df_x["Age"].mean(),inplace = True)
df_x['Age']
```

0	22.000000
1	38.000000
2	26.000000
3	35.000000
4	35.000000
...	
886	27.000000
887	19.000000
888	29.699118
889	26.000000
890	32.000000

Name: Age, Length: 891, dtype: float64

## LabelEncoder

### Encoding District:

```
# we encoded the District column with label encoder to New District
# this will assign numeric values by dictionary order in New District
from sklearn.preprocessing import LabelEncoder

lb = LabelEncoder()
df["New District"] = lb.fit_transform(df["District"])
df
```

	District	Size	Population	Speciality	New District
0	Dhaka	1432	2250000	Administrative	6
1	Gazipur	879	567984	Industry	9
2	Narayanganj	576	53426	Industry	11
3	Rajbari	897	65899	Agriculture	12
4	Chittagong	1234	1345566	Industry	3
5	Cox's Bazar	456	46567	Tourist	5
6	Bandarban	345	67579	Tourist	0
7	Khustia	432	57798	Business	10
8	Feni	543	67890	Agriculture	8
9	Comilla	564	77898	Tourist	4
10	Barisal	577	89750	Agriculture	1
11	Faridpur	567	77650	Agriculture	7
12	Rangpur	575	78966	Agriculture	13
13	Chadpur	876	67789	Tourist	2

## Encoding Speciality:

```
# We can encode the speciality column the same way
df["New Speciality"] = lb.fit_transform(df["Speciality"])
df
```

	District	Size	Population	Speciality	New District	New Speciality
0	Dhaka	1432	2250000	Administrative	6	0
1	Gazipur	879	567984	Industry	9	3
2	Narayanganj	576	53426	Industry	11	3
3	Rajbari	897	65899	Agriculture	12	1
4	Chittagong	1234	1345566	Industry	3	3
5	Cox's Bazer	456	46567	Tourist	5	4
6	Bandarban	345	67579	Tourist	0	4
7	Khustia	432	57798	Business	10	2
8	Feni	543	67890	Agriculture	8	1
9	Comilla	564	77898	Tourist	4	4
10	Barisal	577	89750	Agriculture	1	1
11	Faridpur	567	77650	Agriculture	7	1
12	Rangpur	575	78966	Agriculture	13	1
13	Chadpur	876	67789	Tourist	2	4

## OneHotEncoder

### Setting up:

```
#we import and set parameters of OneHotEncoder
# sparse=False will return a dense array instead of sparse
#handle_unknown parameter will raise error in unknown detected
#drop = first will drop the first category
from sklearn.preprocessing import OneHotEncoder
oh = OneHotEncoder(sparse=False,
                   handle_unknown='error',
                   drop = 'first')
```

## Encoding District:

```
# Encodes the column to array with binary values
dfn = oh.fit_transform(df[['District']])
dfn

/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encod
warnings.warn(
array([[0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0.],
       [0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],
       [0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1.],
       [0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]])
```

## Creating new DataFrame:

```
#Creating a new dataframe with the OneHotEncoded district data
ohendf = pd.DataFrame(dfn)
ohendf
```

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
4	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
13	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0



## Joining DataFrame:

```
#Here we join both df and ohendf dataframe and put it in df dataframe  
df=df.join(ohendf)  
df
```

	District	Size	Population	Speciality	New District	New Speciality	0	1	2	3	4	5	6	7	8	9	10	11	12
0	Dhaka	1432	2250000	Administrative	6		0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
1	Gazipur	879	567984	Industry	9		3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
2	Narayanganj	576	53426	Industry	11		3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
3	Rajbari	897	65899	Agriculture	12		1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
4	Chittagong	1234	1345566	Industry	3		3	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	Cox's Bazer	456	46567	Tourist	5		4	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	Bandarban	345	67579	Tourist	0		4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	Khustia	432	57798	Business	10		2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
8	Feni	543	67890	Agriculture	8		1	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
9	Comilla	564	77898	Tourist	4		4	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	Barisal	577	89750	Agriculture	1		1	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	Faridpur	567	77650	Agriculture	7		1	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
12	Rangpur	575	78966	Agriculture	13		1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
13	Chadpur	876	67789	Tourist	2		4	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0