

NATIONAL UNIVERSITY OF MODERN LANGUAGES
ISLAMABAD



Artificial Neural Network

Lab Quiz: 02

Submitted to
Dr. Moiz-ullah Ghouri

Submitted By
Junaid Asif
(BSAI-144)

Submission Date: December 03, 2024

Lab Quiz

🔗 Generate + Code + Markdown

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')

test_data = pd.read_csv('test_Titanic.csv')
test_data.sample(5)
```

	PassengerId	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
280	1172	3	Oreskovic, Miss. Jelka	female	23.0	0	0	315085	8.6625	NaN	S
132	1024	3	Lefebre, Mrs. Frank (Frances)	female	NaN	0	4	4133	25.4667	NaN	S
20	912	1	Rothschild, Mr. Martin	male	55.0	1	0	PC 17603	59.4000	NaN	C
65	957	2	Corey, Mrs. Percy C (Mary Phyllis Elizabeth Mi...	female	NaN	0	0	F.C.C. 13534	21.0000	NaN	S
269	1161	3	Pokrnic, Mr. Mate	male	17.0	0	0	315095	8.6625	NaN	S

```
test_data.set_index('PassengerId', inplace=True)
test_data.sample(5)
```

	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
PassengerId										
1184	3	Nasr, Mr. Mustafa	male	NaN	0	0	2652	7.2292	NaN	C
1195	3	Pokrnic, Mr. Tome	male	24.0	0	0	315092	8.6625	NaN	S
1054	2	Wright, Miss. Marion	female	26.0	0	0	220844	13.5000	NaN	S
1146	3	Wenzel, Mr. Linhart	male	32.5	0	0	345775	9.5000	NaN	S
976	2	Lamb, Mr. John Joseph	male	NaN	0	0	240261	10.7083	NaN	Q

```
def calc_missing_values(df):
    missing_count = df.isnull().sum().sort_values(ascending=False)
    missing_percent = round(missing_count / len(df) * 100, 2)
    missing_values = pd.concat([missing_count, missing_percent], axis=1, keys=['Count', 'Percent'])
    return missing_values

calc_missing_values(test_data)
```

	Count	Percent
Cabin	327	78.23
Age	86	20.57
Fare	1	0.24
Name	0	0.00
Pclass	0	0.00
PassengerId	0	0.00
Gender	0	0.00
Parch	0	0.00
SibSp	0	0.00
Ticket	0	0.00
Embarked	0	0.00

```
print('Mean age of passengers:', test_data.Age.mean())
print('Median age of passengers:', test_data.Age.median())
```

Mean age of passengers: 30.272590361445783
Median age of passengers: 27.0

```
test_data.Age.fillna(test_data.Age.mean(), inplace=True)
```

```
print('Mode of Embarked is:', test_data.Embarked.mode().values[0])
```

Mode of Embarked is: S

```
test_data.Embarked.fillna(test_data.Embarked.mode().values[0], inplace=True)
```

```
print('Mean fare of passengers:', test_data.Fare.mean())
print('Median fare of passengers:', test_data.Fare.median())
```

Mean fare of passengers: 35.627188489208635
Median fare of passengers: 14.4542

```
test_data.Fare.fillna(test_data.Fare.mean(), inplace=True)
```

```
columns_to_drop = ['Name', 'Ticket', 'Cabin']
test_data.drop(columns=columns_to_drop, axis=1, inplace=True)
```

```
test_data.head()
```

	Pclass	Gender	Age	SibSp	Parch	Fare	Embarked
PassengerId							
892	3	male	34.5	0	0	7.8292	Q
893	3	female	47.0	1	0	7.0000	S
894	2	male	62.0	0	0	9.6875	Q
895	3	male	27.0	0	0	8.6625	S
896	3	female	22.0	1	1	12.2875	S

```
test_data = pd.get_dummies(test_data, columns=['Gender', 'Embarked'], drop_first=True)
```

```
test_data.head()
```

	Pclass	Age	SibSp	Parch	Fare	Gender_male	Embarked_Q	Embarked_S
PassengerId								
892	3	34.5	0	0	7.8292	True	True	False
893	3	47.0	1	0	7.0000	False	False	True
894	2	62.0	0	0	9.6875	True	True	False
895	3	27.0	0	0	8.6625	True	False	True
896	3	22.0	1	1	12.2875	False	False	True

```
def Bucketize_Age(age):
    if age <= 12:
        return 0
    elif age <= 19:
        return 1
    elif age <= 59:
        return 2
    else:
        return 3
test_data['Age'] = test_data['Age'].apply(Bucketize_Age)
print(test_data.head())
```

```

Pclass Age SibSp Parch Fare Gender_male Embarked_Q \
PassengerId
892      3   2     0     0  7.8292         True         True
893      3   2     1     0  7.0000        False        False
894      2   3     0     0  9.6875         True         True
895      3   2     0     0  8.6625         True        False
896      3   2     1     1 12.2875        False        False

Embarked_S
PassengerId
892      False
893       True
894      False
895       True
896       True

#Now replace true and false with 1 and 0
test_data['Gender_male'] = test_data['Gender_male'].apply(lambda x: 1 if x == True else 0)
test_data['Embarked_Q'] = test_data['Embarked_Q'].apply(lambda x: 1 if x == True else 0)
test_data['Embarked_S'] = test_data['Embarked_S'].apply(lambda x: 1 if x == True else 0)

```

```
test_data.head()
```

	Pclass	Age	SibSp	Parch	Fare	Gender_male	Embarked_Q	Embarked_S
PassengerId								
892	3	2	0	0	7.8292	1	1	0
893	3	2	1	0	7.0000	0	0	1
894	2	3	0	0	9.6875	1	1	0
895	3	2	0	0	8.6625	1	0	1
896	3	2	1	1	12.2875	0	0	1

```

first_quartile_fare = test_data.Fare.quantile(0.25)
second_quartile_fare = test_data.Fare.quantile(0.50)
third_quartile_fare = test_data.Fare.quantile(0.75)
def fare_bucketizer(r):
    if r <= first_quartile_fare:
        return 0
    elif r <= second_quartile_fare:
        return 1
    elif r <= third_quartile_fare:
        return 2
    else:
        return 3
test_data['Fare'] = test_data['Fare'].apply(fare_bucketizer)

```

```
test_data.head()
```

	Pclass	Age	SibSp	Parch	Fare	Gender_male	Embarked_Q	Embarked_S
PassengerId								
892	3	2	0	0	0	1	1	0
893	3	2	1	0	0	0	0	1
894	2	3	0	0	1	1	1	0
895	3	2	0	0	1	1	0	1
896	3	2	1	1	1	0	0	1

```
test_data.to_csv('test_cleaned_data.csv')
```

```
clean_data = pd.read_csv('test_cleaned_data.csv')
```

```
clean_data.head()
```

	PassengerId	Pclass	Age	SibSp	Parch	Fare	Gender_male	Embarked_Q	Embarked_S
0	892	3	2	0	0	0	1	1	0
1	893	3	2	1	0	0	0	0	1
2	894	2	3	0	0	1	1	1	0
3	895	3	2	0	0	1	1	0	1
4	896	3	2	1	1	1	0	0	1

