Data Mining and Web algorithm

Lab Assignment 3:

[28 Feb – 5 Mar, 2022]

Patil Amit Gurusidhappa
19104004
B11

Q1: Consider a dataset titanic.csv having attribute such as PassengerId, Pclass, Name, Sex,
Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked, Survived.
Do the following questions: (using python)

```
import pandas as pd
import numpy as np
```

1. Import the Titanic.csv dataset.

```
df=pd.read_csv('E:/Work/JIIT/sem_6/JIIT-SEM-6/DataMining&WebAlgorithms/lab3_dmwa/titanic_dataset.csv');
print(df)
```

```
...  Output exceeds the size limit. Open the full output data in a text editor
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           ...       ...     ...
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         0       3
```

2. Display the complete detailed statistics about the import dataset.

```
# q2
stat=df.describe();
print(stat);
```

```
...        PassengerId    Survived      Pclass         Age       SibSp  \
count   891.000000  891.000000  891.000000  714.000000  891.000000
mean    446.000000    0.383838    2.308642   29.699118    0.523008
std     257.353842    0.486592    0.836071   14.526497    1.102743
min       1.000000    0.000000    1.000000    0.420000    0.000000
25%     223.500000    0.000000    2.000000   20.125000    0.000000
50%     446.000000    0.000000    3.000000   28.000000    0.000000
75%     668.500000    1.000000    3.000000   38.000000    1.000000
max     891.000000    1.000000    3.000000   80.000000    8.000000

             Parch        Fare
count  891.000000  891.000000
mean     0.381594   32.204208
std      0.806057   49.693429
min      0.000000    0.000000
25%      0.000000    7.910400
50%      0.000000   14.454200
75%      0.000000   31.000000
max      6.000000  512.329200
```

3. Display the first ten and last ten records of the dataset.

```
df.head(n=10)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	F
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1

LEMS 1 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER powershell

rsion 22.0.3 is available.

```
df.tail(10)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
881	882	0	3	Markun, Mr. Johann	male	33.0	0	0	349257
882	883	0	3	Dahlberg, Miss. Gerda Ulrika	female	22.0	0	0	7552
883	884	0	2	Banfield, Mr. Frederick James	male	28.0	0	0	C.A./SOTON 34068
884	885	0	3	Sutehall, Mr. Henry Jr	male	25.0	0	0	SOTON/OQ 392076
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652

BLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER powershell

4. Find the missing values in each of the attributes.

```
# q4
df.isna()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cab
0	False	False	False	False	False	False	False	False	False	False	Tr
1	False	False	False	False	False	False	False	False	False	False	Fa
2	False	False	False	False	False	False	False	False	False	False	Tr
3	False	False	False	False	False	False	False	False	False	False	Fa
4	False	False	False	False	False	False	False	False	False	False	Tr
...	...	...	...	...	...	...	...	...	...	...	
886	False	False	False	False	False	False	False	False	False	False	Tr
887	False	False	False	False	False	False	False	False	False	False	Fa
888	False	False	False	False	False	True	False	False	False	False	Tr
889	False	False	False	False	False	False	False	False	False	False	Fa
890	False	False	False	False	False	False	False	False	False	False	Tr

891 rows × 12 columns

5. Replace missing values of one attribute with mean value.

```
# q5
df['Age'].fillna(df['Age'].mean())
```

```
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
```

6. Delete the missing values record for another attribute.

```
# 6
```

```
df.dropna(subset=['Pclass'])
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	7
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	5
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	

7. Try to replace the value of an attribute with a fixed value like “-40”.
8. Find the count of passengers’ gender wise.

```
# 8
male=df['Sex'].value_counts()["male"]
female=df['Sex'].value_counts()["female"]
print(male)
print(female)
```

```
577
314
```

9. Display the count of male senior citizens on titanic.

```
# 9
seniourCitizens=df[df['Age']>70]
```

```
maleSeniourCitizens=seniourCitizens['Sex'].value_counts()["male"]
maleSeniourCitizens
```

```
5
```

10. Find the average fare paid by female passengers.

```
# 10
femalePassengers=df[df['Sex']=="female"]
avgOffemalePassengers=femalePassengers["Fare"].mean()
avgOffemalePassengers
```

```
0.6s
44.47981783439491
```

11. Find the total passengers on titanic who survived.

```
# 11
survived=df['Survived'].value_counts()[1]
survived
```

```
342
```

12. Find the correlation of survived with age and fare.

13. Replace all the gender Males with 1 and Females with 0.

```
# 13
df1=df.replace({'Sex':'male'},1);
df1=df.replace({'Sex':'female'},0);
df1.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25(
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	38.0	1	0	PC 17599	71.283
2	3	1	3	Heikkinen, Miss. Laina	0	26.0	0	0	STON/O2. 3101282	7.925
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	35.0	1	0	113803	53.10(
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05(

14. Find the correlation of survived with gender.

```
# 14
female=df[df['Sex']=='female']
male=df[df['Sex']=='male']
femalesurvived=female['Survived'].value_counts()[1]
print(f'Female Survived : {femalesurvived}')
# malesurvived=male['Survived'].value_counts()[1]
# malesurvived
# male
```

```
Female Survived : 233
```

15. If the name of the dataframe consisting the titanic.csv dataset is df then what will be the
outcome of the following commands:

15.1

```
df.groupby('Sex').Age.median()
```

[31] ✓ 0.7s Python

```
Sex
female    27.0
male      29.0
Name: Age, dtype: float64
```

15.2

```
df.groupby('Pclass').agg({'Fare' : 'mean','Age':'median'})
```

[32] ✓ 0.7s Python

Pclass	Fare	Age
1	84.154687	37.0
2	20.662183	29.0
3	13.675550	24.0

15.3

```
df.Pclass.value_counts()
```

[33] ✓ 0.6s Python

```
3    491
1    216
2    184
Name: Pclass, dtype: int64
```

15.4

```
df.describe(include='all')
```

✓ 0.1s Python

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	
count	891.000000	891.000000	891.000000	891	891	714.000000	891.000000	891.(
unique	NaN	NaN	NaN	891	2	NaN	NaN	
top	NaN	NaN	NaN	Braund, Mr. Owen Harris	male	NaN	NaN	
freq	NaN	NaN	NaN	1	577	NaN	NaN	
mean	446.000000	0.383838	2.308642	NaN	NaN	29.699118	0.523008	0.:
std	257.353842	0.486592	0.836071	NaN	NaN	14.526497	1.102743	0.(
min	1.000000	0.000000	1.000000	NaN	NaN	0.420000	0.000000	0.(
25%	223.500000	0.000000	2.000000	NaN	NaN	20.125000	0.000000	0.(
50%	446.000000	0.000000	3.000000	NaN	NaN	28.000000	0.000000	0.(

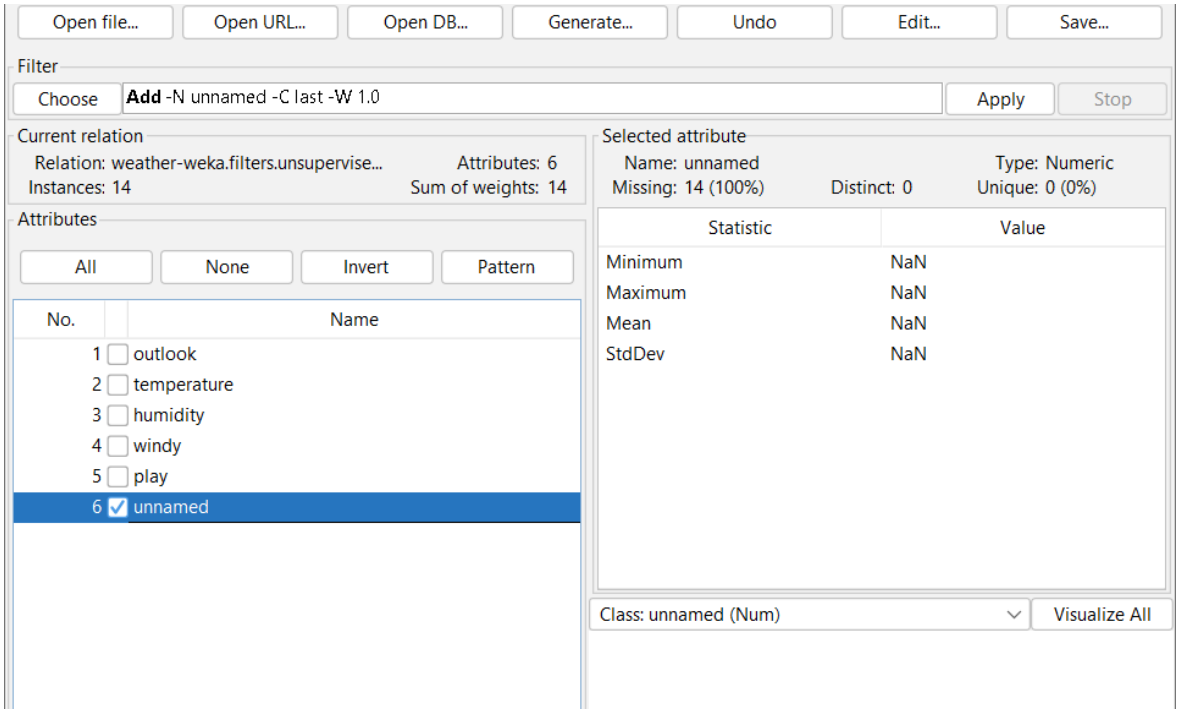
15.5

```
df.loc[((df.Age> 30) & (df.Sex == 'male')), 'Name':]
```

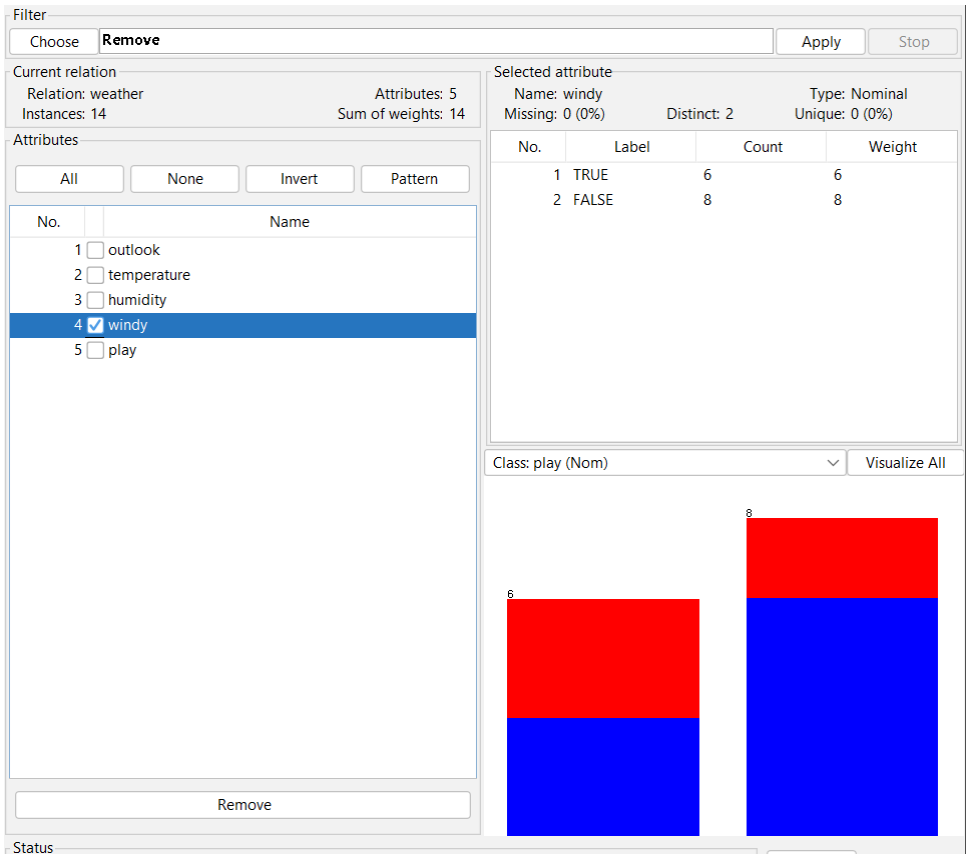
✓ 0.6s Python

	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
4	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
6	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
13	Andersson, Mr. Anders Johan	male	39.0	1	5	347082	31.2750	NaN	S
20	Fynney, Mr. Joseph J	male	35.0	0	0	239865	26.0000	NaN	S
21	Beesley, Mr. Lawrence	male	34.0	0	0	248698	13.0000	D56	S
...	...	...	...	...	...	...	...	...	...
867	Roebling, Mr. Washington Augustus II	male	31.0	0	0	PC 17590	50.4958	A24	S
872	Carlsson, Mr. Frans Olof	male	33.0	0	0	695	5.0000	B51 B53	S

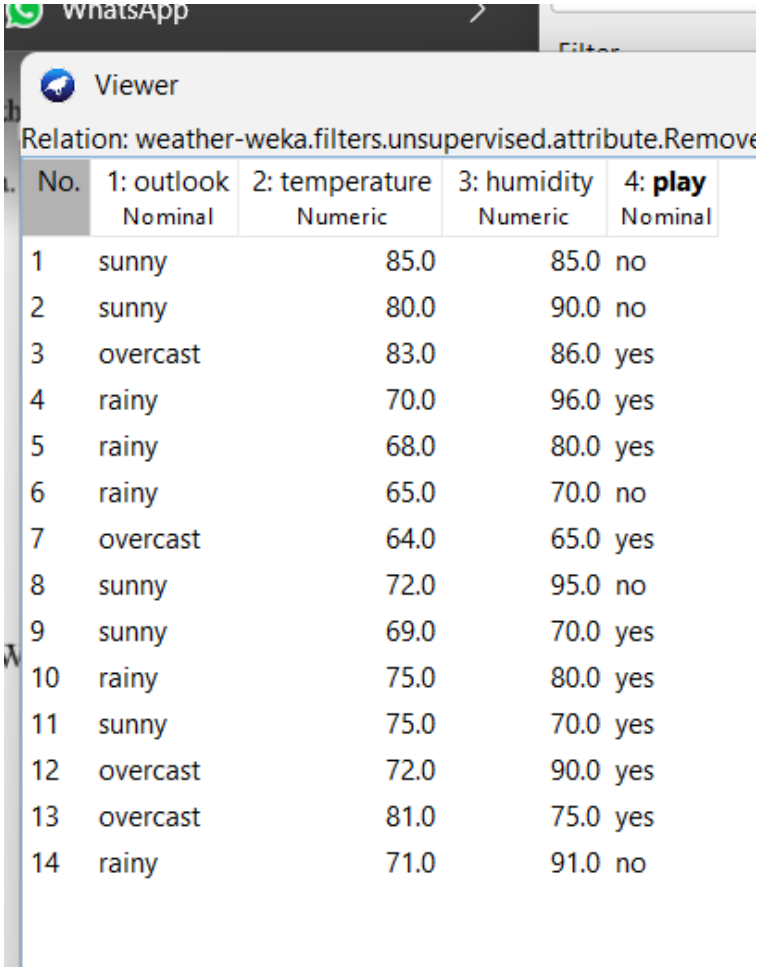
Q2.A


Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
Add -N unnamed -C last -W 1.0
Apply
Stop
Current relation
Relation: weather-weka.filters.unsupervise...
Attributes: 6
Instances: 14
Sum of weights: 14
Selected attribute
Name: unnamed
Type: Numeric
Missing: 14 (100%)
Distinct: 0
Unique: 0 (0%)
Attributes
All
None
Invert
Pattern
No.
Name
1 outlook
2 temperature
3 humidity
4 windy
5 play
6 unnamed
Statistic
Value
Minimum
NaN
Maximum
NaN
Mean
NaN
StdDev
NaN
Class: unnamed (Num)
Visualize All

Q2.B


Filter
Choose
Remove
Apply
Stop
Current relation
Relation: weather
Attributes: 5
Instances: 14
Sum of weights: 14
Selected attribute
Name: windy
Type: Nominal
Missing: 0 (0%)
Distinct: 2
Unique: 0 (0%)
Attributes
All
None
Invert
Pattern
No.
Name
1 outlook
2 temperature
3 humidity
4 windy
5 play
No.
Label
Count
Weight
1 TRUE
6
6
2 FALSE
8
8
Class: play (Nom)
Visualize All
8
6
Remove
Status

After Removal



Viewer

Relation: weather-weka.filters.unsupervised.attribute.Remove

No.	1: outlook Nominal	2: temperature Numeric	3: humidity Numeric	4: **play** Nominal
1	sunny	85.0	85.0	no
2	sunny	80.0	90.0	no
3	overcast	83.0	86.0	yes
4	rainy	70.0	96.0	yes
5	rainy	68.0	80.0	yes
6	rainy	65.0	70.0	no
7	overcast	64.0	65.0	yes
8	sunny	72.0	95.0	no
9	sunny	69.0	70.0	yes
10	rainy	75.0	80.0	yes
11	sunny	75.0	70.0	yes
12	overcast	72.0	90.0	yes
13	overcast	81.0	75.0	yes
14	rainy	71.0	91.0	no

Q2.C