

Machine Learning Lab

Exercise 4 (Week 6):

Pandas

1. Copy file abc.xlsx locally.
2. Use following url for csv (comma separated)file:
 - <https://tinyurl.com/titanic-csv>
3. Use following url for csv (semicolon separated) file:
 - <https://tinyurl.com/yx3b6sq3>
4. Use files mentioned in 1 or 2 or 3, for exercises given below.
5. Dimension of Dataframes
 - `print(data.shape)`
6. selecting specific rows
 - `print(data.iloc[4])`
7. Slicing of rows
 - `print(data.iloc[1:4])`
8. Selecting all rows up to a point
 - `print(data.iloc[:4])`
 - Select one to another columns:
 - `print(data.loc[1:3,'Name':'Survived'])`
9. Selecting Rows based on conditions:
`import pandas as pd`
`url='https://tinyurl.com/yx3b6sq3'`
`data=pd.read_csv(url,sep=';',header=None,prefix='Column')`
`print(data[(data['Column1']==0.580) & (data['Column4']==1)])`
10. Replacing Values in a dataframe
`d1=data["PClass"].replace("1st","First")`
`print(data)`
`print(d1)`
`d2=data["Sex"].replace(["female","male"],["Woman","Man"]).head()`
`print(data)`
`print(d2)`
11. Inplace data replacement:
`data['Sex'].replace(['female','male'],['Woman','Man'],inplace=True)`
`print(data)`
12. Renaming Columns;
`d3=data.rename(columns={'Sex':'Gender'})`
`print(d3.head())`
`data.rename(columns={'Sex':'Gender'},inplace=True)`

```
print(data.head())
```

It is also possible to use multi column rename using dictionary

13. finding Min, Max, Count, Avg, and Sum

```
print(data['Age'].mean())
```

and likewise

Use following function on dataset columns

Variance - var()

Standard Deviation - std()

Kurtosis - kurt()

Skewness - skew()

Std error of the mean - sem()

Mode - mode()

Median- median()

14. Unique values and count of it

```
print(data['PClass'].unique)
```

```
print(data['Age'].value_counts())
```

```
print(data['PClass'].value_counts())
```

15. Dropping a column from dataframes

```
d1=data.drop('Age',axis=1)
```

```
import numpy as np
```

```
pd.set_option('display.max_columns', None)
```

```
pd.set_option('display.width', None)
```

```
print(d1.head())
```

```
print(data.head())
```

```
data.drop('Age',axis=1,inplace=True)
```

```
print(data)
```

16. Dropping multiple columns

```
d1=data.drop(['Age','PClass'],axis=1).head()
```

17. Dropping a column not having names;

```
d1=data.drop(data.columns[1],axis=1).head()
```

18. Deleting rows

```
print(data[data['Age']>30])
```

```
d1=data[data['Age']>30]
```

```
print(d1)
```

19. Deleting duplicate rows

```
print(data.drop_duplicates())
```

```
print(data)
```

```
d1=data.drop_duplicates()
```

```
print(d1)
```

```
d2=data.drop_duplicates(subset=['Maths'])
```

```
print(d2)
```

20. Grouping data and analysing

```
print(data.groupby('Survived').count())
```

21. Concatenation of Data set (Row wise)

```
dataa=pd.read_excel(r'C:\Users\Netraa\Desktop\abc.xlsx',sheet_name=0)
datab=pd.read_excel(r'C:\Users\Netraa\Desktop\abc.xlsx',sheet_name=1)
datac=pd.concat([dataa,datab],axis=0)
print(datac)
```

22. Concatenation of Data set (Column wise)

```
datac=pd.concat([dataa,datab],axis=1)
please make not of axis argument
```

23. Merging Data Frames (create Excel files with for EMP and COMP as we used in subject of Database)

inner join

```
pd.merge(EMP,COMP,on='CID')
```

outer joins

```
pd.merge((EMP,COMP,on='CID',how='left') #use 'outer' or 'right'
instead of 'left')
```