Lab # 2

Dataset Preparation with Excel spreadsheet and Dataset Preprocessing and Scaling techniques

**OBJECTIVE**

Dataset preparation by selecting all the possible features on the given scenario using excel. Load designated data set to working environment. Checking the data set for missing values and outliers. Implementing Normalization and Standardization techniques to scale the values.

1. **Pandas**

Pandas is a very powerful and scalable tool for data analysis. It supports multiple file format as we might get the data in any format. Pandas also have support for excel file format.

We first need to import Pandas and load excel file, and then parse excel file sheets as a Pandas data frame. It can be done using,

import pandas as pd

# Import the excel file in python environment

excel\_file = pd.ExcelFile('file\_1.xlsx') # you can use pd.read\_excel too.

We can even see the sheet that we are working on using,

print(excel\_file.sheet\_names)

In order to load the sheet in the data frame, we can use,

df = excel\_file.parse('Sheet1')

print(df)

1. **Pandas Data Frame.**

Let’s see how to read excel files to Pandas data frame objects using **Pandas**.

**Example 1**: Read an excel file using read\_excel() method of pandas.

# import pandas lib as pd

import pandas as pd

# read by default 1st sheet of an excel file

dataframe1 = pd.read\_excel('SampleWork.xlsx')

print(dataframe1)

**Output:**

Name Age Stream Percentage

0 Ali 30 Math 65

1 Faizan 39 Science 90

2 Abrar 44 Commerce 75

3 Hareem 23 Math 85

4 Kashif 21 Science 70

It is possible to load a desired sheet by simply introducing,

dataframe2 = pd.read\_excel('SampleWork.xlsx', sheet\_name = 1) # this instruction will load only sheet 1. If sheet\_name is selected to be None, it will print all the sheets. Skipows is also a function that can be placed after the sheet\_name to skip the desired number of rows from the excel work spreadsheet.

print(dataframe2)

If we require specific features for our Machine learning algorithm, we can select these required columns as well by simply using,

require\_cols = [0, 3]

required\_df = pd.read\_excel('SampleWork.xlsx', usecols = require\_cols)

print(required\_df)

**Output**

Name Percentage

0 Ankit 95

1 Rahul 90

2 Shaurya 85

3 Aishwarya 80

4 Priyanka 75

We can create a pandas data frame by using a following code,

|  |
| --- |
| # importing pandas as pd  import pandas as pd    # Creating the dataframe  df = pd.DataFrame({"A":[1, 5, 3, 4, 2],                     "B":[3, 2, 4, 3, 4],                     "C":[2, 2, 7, 3, 4],                     "D":[4, 3, 6, 12, 7]},                     index =["A1", "A2", "A3", "A4", "A5"])  df  **output**  A screenshot of a computer  Description automatically generated with low confidence |

## WORKING WITH MISSING DATA:

Missing Data can occur when no information is provided for one or more items or for a whole unit. Missing Data is a very big problem in real life scenario. Missing Data can also refer to as NA (Not Available) values in pandas.

**Checking for missing values using isnull()**

In order to check null values in Pandas DataFrame, we use isnull() function, this function returns data frame of Boolean values which are True for NaN values.

*# importing pandas as pd*

**import** pandas **as** pd

*# importing numpy as np*

**import** numpy **as** np

*# dictionary of lists*

dict = {**'First Score'**: [100, 90, np.nan, 95],

**'Second Score'**: [30, 45, 56, np.nan],

**'Third Score'**: [np.nan, 40, 80, 98]}

*# creating a dataframe from list*

df = pd.DataFrame(dict)

*# using isnull() function*

print(df.isnull())

Table

Description automatically generated with medium confidence

**Checking for missing values using notnull()**

In order to check null values in Pandas Dataframe, we use notnull() function, this function returns dataframe of Boolean values which are False for NaN values.

*# importing pandas as pd*

**import** pandas **as** pd

*# importing numpy as np*

**import** numpy **as** np

*# dictionary of lists*

dict = {**'First Score'**: [100, 90, np.nan, 95],

**'Second Score'**: [30, 45, 56, np.nan],

**'Third Score'**: [np.nan, 40, 80, 98]}

*# creating a dataframe using dictionary*

df = pd.DataFrame(dict)

*# using notnull() function*

print(df.notnull())

Graphical user interface, table

Description automatically generated

**Filling missing values using fillna(), replace() and interpolate()**

In order to fill null values in datasets, we use fillna(), replace() and interpolate() functions, these functions replace NaN values with some value of their own. All these functions help in filling null values in datasets of a DataFrame. Interpolate() function is basically used to fill NA values in the dataframe but it uses various interpolation techniques to fill the missing values rather than hard-coding the values.

*# importing pandas as pd*

**import** pandas **as** pd

*# importing numpy as np*

**import** numpy **as** np

*# dictionary of lists*

dict = {**'First Score'**: [100, 90, np.nan, 95],

**'Second Score'**: [30, 45, 56, np.nan],

**'Third Score'**: [np.nan, 40, 80, 98]}

*# creating a dataframe from dictionary*

df = pd.DataFrame(dict)

*# filling missing value using fillna()*

print(df.fillna(0))

Table

Description automatically generated

*# filling missing values with the previous ones using fillna()*

print(df.fillna(method=**'pad'**))

Table

Description automatically generated

*# filling missing values with the next ones using fillna()*

print(df.fillna(method=**'bfill'**))

Table

Description automatically generated

*# will replace Nan value in dataframe with value 0*

print(df.replace(to\_replace=np.nan, value=0))

*# to interpolate the missing values*

print(df.interpolate(method=**'linear'**, limit\_direction=**'forward'**))

## OUTLIERS

Table

Description automatically generated

An **outlier** is an observation point that is distant from other observations.

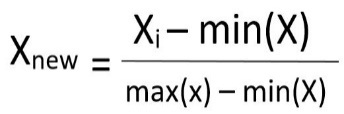
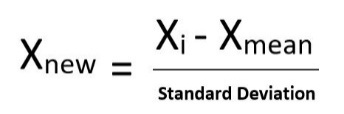
As you can see from the above collected data that all other players scored 300+ except Player3 who scored 10. This figure can be just a typing **mistake**or it is showing the **variance** in your data and indicating that Player3 is performing very bad so, needs improvements. In case of a mistake, we can simply interpolate that value.

## FEATURE SCALING USING NORMALIZATION AND STANDARDIZATION:

Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. It is performed during the data pre-processing to handle highly varying magnitudes or values or units. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values.

**Example:** If an algorithm is not using feature scaling method then it can consider the value 3000 meter to be greater than 5 km but that’s actually not true and in this case, the algorithm will give wrong predictions. So, we use Feature Scaling to bring all values to same magnitudes and thus, tackle this issue.

**Techniques to perform Feature Scaling**

* Min-Max Normalization: This technique re-scales a feature or observation value with distribution value between 0 and 1.  
  
* Standardization: It is a very effective technique which re-scales a feature value so that it has distribution with 0 mean value and variance equals to 1.  
  

**CODE: (Scaling through Min-Max Normalization)**

import numpy as np

from sklearn import preprocessing

x = np.array([[1,2,3],[4,5,6],[7,8,9]])

minmax = preprocessing.MinMaxScaler(feature\_range=(0,1))

print(minmax.fit(x).transform(x))

**OUTPUT:**

A picture containing graphical user interface

Description automatically generated

**CODE: (Scaling through Standardization)**

import numpy as np

from sklearn import preprocessing

x = np.array([[1,2,3],[4,5,6],[7,8,9]])

standard = preprocessing.StandardScaler().fit(x)

print(standard.transform(x))

**OUTPUT:**

Text

Description automatically generated with low confidence

**Lab Tasks:**

1. Write a python code to load an excel spreadsheet containing two different sheets and print both of them.
2. Write a python cade to generate a pandas data frame having 4 columns and 5 rows. Column 1 must contain the index values like Ali, Amir, Kamran, etc and Row 1 must contain the subject names.
3. Write a python code to read an excel spreadsheet and only print first two columns using pandas data frame.
4. Write a python code to skip the first two rows of excel spreadsheet and print the output using pandas data frame.
5. Write a python code to fill all the null values in Gender column of employees.csv with “No Gender”. Print the first 10 to 30 rows of the data frame for visualization.
6. Write a python code to scale the values of features (Age and Salary) using Min-Max Normalization technique. Verify your answers by applying the formula mentioned above.
7. Write a python code to scale the values of features (Age and Salary) using Standardization technique. Verify your answers by applying the formula mentioned above.

Table

Description automatically generated

1. Given this dictionary, create a dataframe from dictionary and interpolate the missing values using backward interpolation. Hint: use interpolate().

dict = {**'First Score'**: [100, 90, np.nan, 95],

**'Second Score'**: [30, 45, 56, np.nan],

**'Third Score'**: [np.nan, 40, 80, 98]}