practical1-dsbda

March 4, 2025

0.0.1 *Piyusha Supe(23CO315)*

0.1 Data Wrangling 1

Perform the following operations using Python on any open source dataset (e.g., data.csv) 1. Import all the required Python Libraries. 2. Locate an open source data from the web (e.g., https://www.kaggle.com). Provide a clear description of the data and its source (i.e., URL of the web site). 3. Load the Dataset into pandas dataframe. 4. Data Preprocessing: check for missing values in the data using pandas isnull(), describe() function to get some initial statistics. Provide variable descriptions. Types of variables etc. Check the dimensions of the data frame. 5. Data Formatting and Data Normalization: Summarize the types of variables by checking the data types (i.e., character, numeric, integer, factor, and logical) of the variables in the data set. If variables are not in the correct data type, apply proper type conversions. Turn categorical variables into quantitative variables in Python. In addition to the codes and outputs, explain every operation that you do in the above steps and explain everything that you do to import/read/scrape the data set

1. Import all the required Python Libraries

[3]:
```
import pandas as pd
import numpy as np
```

2. Locate an open source data from the web (e.g., https://www.kaggle.com). Provide a clear description of the data and its source (i.e., URL of the web site).
3. Load the Dataset into pandas dataframe

[5]:
```
from google.colab import files
 files.upload()
```

```
<IPython.core.display.HTML object>
```

```
Saving IRIS.csv to IRIS.csv
```

[5]:
```
{'IRIS.csv': b'sepal_length,sepal_width,petal_length,petal_width,species\r\n5.1,
3.5,1.4,0.2,Iris-setosa\r\n4.9,3,1.4,0.2,Iris-setosa\r\n4.7,3.2,1.3,0.2,Iris-
setosa\r\n4.6,3.1,1.5,0.2,Iris-setosa\r\n5,3.6,1.4,0.2,Iris-
setosa\r\n5.4,3.9,1.7,0.4,Iris-setosa\r\n4.6,3.4,1.4,0.3,Iris-
setosa\r\n5,3.4,1.5,0.2,Iris-setosa\r\n4.4,2.9,1.4,0.2,Iris-
setosa\r\n4.9,3.1,1.5,0.1,Iris-setosa\r\n5.4,3.7,1.5,0.2,Iris-
setosa\r\n4.8,3.4,1.6,0.2,Iris-setosa\r\n4.8,3,1.4,0.1,Iris-
```

setosa\r\n4.3,3,1.1,0.1,Iris-setosa\r\n5.8,4,1.2,0.2,Iris-
setosa\r\n5.7,4.4,1.5,0.4,Iris-setosa\r\n5.4,3.9,1.3,0.4,Iris-
setosa\r\n5.1,3.5,1.4,0.3,Iris-setosa\r\n5.7,3.8,1.7,0.3,Iris-
setosa\r\n5.1,3.8,1.5,0.3,Iris-setosa\r\n5.4,3.4,1.7,0.2,Iris-
setosa\r\n5.1,3.7,1.5,0.4,Iris-setosa\r\n4.6,3.6,1,0.2,Iris-
setosa\r\n5.1,3.3,1.7,0.5,Iris-setosa\r\n4.8,3.4,1.9,0.2,Iris-
setosa\r\n5,3,1.6,0.2,Iris-setosa\r\n5,3.4,1.6,0.4,Iris-
setosa\r\n5.2,3.5,1.5,0.2,Iris-setosa\r\n5.2,3.4,1.4,0.2,Iris-
setosa\r\n4.7,3.2,1.6,0.2,Iris-setosa\r\n4.8,3.1,1.6,0.2,Iris-
setosa\r\n5.4,3.4,1.5,0.4,Iris-setosa\r\n5.2,4.1,1.5,0.1,Iris-
setosa\r\n5.5,4.2,1.4,0.2,Iris-setosa\r\n4.9,3.1,1.5,0.1,Iris-
setosa\r\n5,3.2,1.2,0.2,Iris-setosa\r\n5.5,3.5,1.3,0.2,Iris-
setosa\r\n4.9,3.1,1.5,0.1,Iris-setosa\r\n4.4,3,1.3,0.2,Iris-
setosa\r\n5.1,3.4,1.5,0.2,Iris-setosa\r\n5,3.5,1.3,0.3,Iris-
setosa\r\n4.5,2.3,1.3,0.3,Iris-setosa\r\n4.4,3.2,1.3,0.2,Iris-
setosa\r\n5,3.5,1.6,0.6,Iris-setosa\r\n5.1,3.8,1.9,0.4,Iris-
setosa\r\n4.8,3,1.4,0.3,Iris-setosa\r\n5.1,3.8,1.6,0.2,Iris-
setosa\r\n4.6,3.2,1.4,0.2,Iris-setosa\r\n5.3,3.7,1.5,0.2,Iris-
setosa\r\n5,3.3,1.4,0.2,Iris-setosa\r\n7,3.2,4.7,1.4,Iris-
versicolor\r\n6.4,3.2,4.5,1.5,Iris-versicolor\r\n6.9,3.1,4.9,1.5,Iris-
versicolor\r\n5.5,2.3,4,1.3,Iris-versicolor\r\n6.5,2.8,4.6,1.5,Iris-
versicolor\r\n5.7,2.8,4.5,1.3,Iris-versicolor\r\n6.3,3.3,4.7,1.6,Iris-
versicolor\r\n4.9,2.4,3.3,1,Iris-versicolor\r\n6.6,2.9,4.6,1.3,Iris-
versicolor\r\n5.2,2.7,3.9,1.4,Iris-versicolor\r\n5,2,3.5,1,Iris-
versicolor\r\n5.9,3,4.2,1.5,Iris-versicolor\r\n6,2.2,4,1,Iris-
versicolor\r\n6.1,2.9,4.7,1.4,Iris-versicolor\r\n5.6,2.9,3.6,1.3,Iris-
versicolor\r\n6.7,3.1,4.4,1.4,Iris-versicolor\r\n5.6,3,4.5,1.5,Iris-
versicolor\r\n5.8,2.7,4.1,1,Iris-versicolor\r\n6.2,2.2,4.5,1.5,Iris-
versicolor\r\n5.6,2.5,3.9,1.1,Iris-versicolor\r\n5.9,3.2,4.8,1.8,Iris-
versicolor\r\n6.1,2.8,4,1.3,Iris-versicolor\r\n6.3,2.5,4.9,1.5,Iris-
versicolor\r\n6.1,2.8,4.7,1.2,Iris-versicolor\r\n6.4,2.9,4.3,1.3,Iris-
versicolor\r\n6.6,3,4.4,1.4,Iris-versicolor\r\n6.8,2.8,4.8,1.4,Iris-
versicolor\r\n6.7,3,5,1.7,Iris-versicolor\r\n6,2.9,4.5,1.5,Iris-
versicolor\r\n5.7,2.6,3.5,1,Iris-versicolor\r\n5.5,2.4,3.8,1.1,Iris-
versicolor\r\n5.5,2.4,3.7,1,Iris-versicolor\r\n5.8,2.7,3.9,1.2,Iris-
versicolor\r\n6,2.7,5.1,1.6,Iris-versicolor\r\n5.4,3,4.5,1.5,Iris-
versicolor\r\n6,3.4,4.5,1.6,Iris-versicolor\r\n6.7,3.1,4.7,1.5,Iris-
versicolor\r\n6.3,2.3,4.4,1.3,Iris-versicolor\r\n5.6,3,4.1,1.3,Iris-
versicolor\r\n5.5,2.5,4,1.3,Iris-versicolor\r\n5.5,2.6,4.4,1.2,Iris-
versicolor\r\n6.1,3,4.6,1.4,Iris-versicolor\r\n5.8,2.6,4,1.2,Iris-
versicolor\r\n5,2.3,3.3,1,Iris-versicolor\r\n5.6,2.7,4.2,1.3,Iris-
versicolor\r\n5.7,3,4.2,1.2,Iris-versicolor\r\n5.7,2.9,4.2,1.3,Iris-
versicolor\r\n6.2,2.9,4.3,1.3,Iris-versicolor\r\n5.1,2.5,3,1.1,Iris-
versicolor\r\n5.7,2.8,4.1,1.3,Iris-versicolor\r\n6.3,3.3,6,2.5,Iris-
virginica\r\n5.8,2.7,5.1,1.9,Iris-virginica\r\n7.1,3,5.9,2.1,Iris-
virginica\r\n6.3,2.9,5.6,1.8,Iris-virginica\r\n6.5,3,5.8,2.2,Iris-
virginica\r\n7.6,3,6.6,2.1,Iris-virginica\r\n4.9,2.5,4.5,1.7,Iris-

```
virginica\r\n7.3,2.9,6.3,1.8,Iris-virginica\r\n6.7,2.5,5.8,1.8,Iris-
virginica\r\n7.2,3.6,6.1,2.5,Iris-virginica\r\n6.5,3.2,5.1,2,Iris-
virginica\r\n6.4,2.7,5.3,1.9,Iris-virginica\r\n6.8,3,5.5,2.1,Iris-
virginica\r\n5.7,2.5,5,2,Iris-virginica\r\n5.8,2.8,5.1,2.4,Iris-
virginica\r\n6.4,3.2,5.3,2.3,Iris-virginica\r\n6.5,3,5.5,1.8,Iris-
virginica\r\n7.7,3.8,6.7,2.2,Iris-virginica\r\n7.7,2.6,6.9,2.3,Iris-
virginica\r\n6,2.2,5,1.5,Iris-virginica\r\n6.9,3.2,5.7,2.3,Iris-
virginica\r\n5.6,2.8,4.9,2,Iris-virginica\r\n7.7,2.8,6.7,2,Iris-
virginica\r\n6.3,2.7,4.9,1.8,Iris-virginica\r\n6.7,3.3,5.7,2.1,Iris-
virginica\r\n7.2,3.2,6,1.8,Iris-virginica\r\n6.2,2.8,4.8,1.8,Iris-
virginica\r\n6.1,3,4.9,1.8,Iris-virginica\r\n6.4,2.8,5.6,2.1,Iris-
virginica\r\n7.2,3,5.8,1.6,Iris-virginica\r\n7.4,2.8,6.1,1.9,Iris-
virginica\r\n7.9,3.8,6.4,2,Iris-virginica\r\n6.4,2.8,5.6,2.2,Iris-
virginica\r\n6.3,2.8,5.1,1.5,Iris-virginica\r\n6.1,2.6,5.6,1.4,Iris-
virginica\r\n7.7,3,6.1,2.3,Iris-virginica\r\n6.3,3.4,5.6,2.4,Iris-
virginica\r\n6.4,3.1,5.5,1.8,Iris-virginica\r\n6,3,4.8,1.8,Iris-
virginica\r\n6.9,3.1,5.4,2.1,Iris-virginica\r\n6.7,3.1,5.6,2.4,Iris-
virginica\r\n6.9,3.1,5.1,2.3,Iris-virginica\r\n5.8,2.7,5.1,1.9,Iris-
virginica\r\n6.8,3.2,5.9,2.3,Iris-virginica\r\n6.7,3.3,5.7,2.5,Iris-
virginica\r\n6.7,3,5.2,2.3,Iris-virginica\r\n6.3,2.5,5,1.9,Iris-
virginica\r\n6.5,3,5.2,2,Iris-virginica\r\n6.2,3.4,5.4,2.3,Iris-
virginica\r\n5.9,3,5.1,1.8,Iris-virginica\r\n'}
```

3.Load the Dataset into pandas dataframe.

[6]:
```
iris=pd.read_csv("/content/IRIS.csv")
```

[7]:
```
iris
```

[7]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
..	…	…	…	…	…
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

[150 rows x 5 columns]

4. Data Preprocessing: check for missing values in the data using pandas isnull(), describe() function to get some initial statistics. Provide variable descriptions. Types of variables etc. Check the dimensions of the data frame

[8]:
```
iris.head()
```

[8]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

[9]:
```
iris.tail()
```

[9]:

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

[10]:
```
iris.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

[11]:
```
iris.describe()
```

[11]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

[12]:
```
iris.describe(include="all")
```

[12]:

	sepal_length	sepal_width	petal_length	petal_width	species
count	150.000000	150.000000	150.000000	150.000000	150
unique	NaN	NaN	NaN	NaN	3
top	NaN	NaN	NaN	NaN	Iris-setosa
freq	NaN	NaN	NaN	NaN	50
mean	5.843333	3.054000	3.758667	1.198667	NaN
std	0.828066	0.433594	1.764420	0.763161	NaN
min	4.300000	2.000000	1.000000	0.100000	NaN
25%	5.100000	2.800000	1.600000	0.300000	NaN
50%	5.800000	3.000000	4.350000	1.300000	NaN
75%	6.400000	3.300000	5.100000	1.800000	NaN
max	7.900000	4.400000	6.900000	2.500000	NaN

5. Data Formatting and Data Normalization: Summarize the types of variables by checking the data types (i.e., character, numeric, integer, factor, and logical) of the variables in the data set. If variables are not in the correct data type, apply proper type conversions

[13]:
```
iris.shape
```

[13]:
```
(150, 5)
```

[14]:
```
iris.size
```

[14]:
```
750
```

[15]:
```
iris.columns
```

[15]:
```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
```

[16]:
```
iris.ndim
```

[16]:
```
2
```

Data Normalization

[17]:
```
iris[0:3]
```

[17]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa

[18]:
```
iris.loc[0:2]
```

[18]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa

```
1           4.9          3.0           1.4          0.2  Iris-setosa
2           4.7          3.2           1.3          0.2  Iris-setosa
```

[19]:
```
iris.loc[0:2,'sepal_length':'petal_width']
```

[19]:
```
   sepal_length  sepal_width  petal_length  petal_width
0           5.1          3.5           1.4          0.2
1           4.9          3.0           1.4          0.2
2           4.7          3.2           1.3          0.2
```

[20]:
```
iris.iloc[1:5,1:5]
```

[20]:
```
   sepal_width  petal_length  petal_width      species
1          3.0           1.4          0.2  Iris-setosa
2          3.2           1.3          0.2  Iris-setosa
3          3.1           1.5          0.2  Iris-setosa
4          3.6           1.4          0.2  Iris-setosa
```

Check if there are any null values

[21]:
```
iris.isnull()
```

[21]:
```
     sepal_length  sepal_width  petal_length  petal_width  species
0           False        False         False        False    False
1           False        False         False        False    False
2           False        False         False        False    False
3           False        False         False        False    False
4           False        False         False        False    False
..            …            …             …            …        …
145         False        False         False        False    False
146         False        False         False        False    False
147         False        False         False        False    False
148         False        False         False        False    False
149         False        False         False        False    False

[150 rows x 5 columns]
```

[ ]:
```
iris.isna()
```

[ ]:
```
     sepal_length  sepal_width  petal_length  petal_width  species
0           False        False         False        False    False
1           False        False         False        False    False
2           False        False         False        False    False
3           False        False         False        False    False
4           False        False         False        False    False
..            …            …             …            …        …
145         False        False         False        False    False
```

```
146         False        False         False        False    False
147         False        False         False        False    False
148         False        False         False        False    False
149         False        False         False        False    False

[150 rows x 5 columns]
```

[ ]:
```
iris.isnull().any()
```

[ ]:
```
sepal_length    False
sepal_width     False
petal_length    False
petal_width     False
species         False
dtype: bool
```

6. Turn categorical variables into quantitative variables in Python. There are many ways to convert categorical data into numerical data. Here the three most used methods are discussed.

i. Label Encoding: Label Encoding refers to converting the labels into a numeric form so as to convert them into the machine-readable form. It is an important preprocessing step for the structured dataset in supervised learning

[ ]:
```
iris.isnull().sum()
```

[ ]:
```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

[ ]:
```
iris.sepal_length.isnull().sum()
```

[ ]:
```
0
```

5. Data Formatting and Data Normalization:

Summarize the types of variables by checking the data types (i.e., character, numeric, integer, factor, and logical) of the variables in the data set. If variables are not in the correct data type, apply proper type conversions

[ ]:
```
iris.dtypes
```

[ ]:
```
sepal_length    float64
sepal_width     float64
petal_length    float64
petal_width     float64
species          object
```

```
dtype: object
```

[ ]:
```
x=iris.iloc[:,:4]
```

[ ]:
```
iris.sepal_length=iris.sepal_length.astype("int")
```

[ ]:
```
iris.dtypes
```

[ ]:
```
sepal_length      int32
sepal_width     float64
petal_length    float64
petal_width     float64
species          object
dtype: object
```

[23]:
```
from sklearn import preprocessing
```

[ ]:
```
iris.head()
```

[ ]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5	3.5	1.4	0.2	Iris-setosa
1	4	3.0	1.4	0.2	Iris-setosa
2	4	3.2	1.3	0.2	Iris-setosa
3	4	3.1	1.5	0.2	Iris-setosa
4	5	3.6	1.4	0.2	Iris-setosa

[24]:
```
min_max_scalar = preprocessing.MinMaxScaler()
```

[26]:
```
x = iris.iloc[:,:4]
x
```

[26]:

	sepal_length	sepal_width	petal_length	petal_width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
..	…	…	…	…
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

[150 rows x 4 columns]

[ ]:
```
x_scaled = min_max_scalar.fit_transform(x)
```

[ ]:
```
df_normalized = pd.DataFrame(x_scaled)
```

[ ]:
```
df_normalized
```

[ ]:
```
             0         1         2         3
0     0.333333  0.625000  0.067797  0.041667
1     0.000000  0.416667  0.067797  0.041667
2     0.000000  0.500000  0.050847  0.041667
3     0.000000  0.458333  0.084746  0.041667
4     0.333333  0.666667  0.067797  0.041667
..         …         …         …         …
145   0.666667  0.416667  0.711864  0.916667
146   0.666667  0.208333  0.677966  0.750000
147   0.666667  0.416667  0.711864  0.791667
148   0.666667  0.583333  0.745763  0.916667
149   0.333333  0.416667  0.694915  0.708333

[150 rows x 4 columns]
```

6. Turn categorical variables into quantitative variables in Python.

i. Label Encoding

[27]:
```
from sklearn import preprocessing
```

[28]:
```
label_encoder =preprocessing.LabelEncoder()
```

[29]:
```
iris['species'].unique()
```

[29]:
```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

[30]:
```
iris['species']=label_encoder.fit_transform(iris['species'])
```

[32]:
```
iris['species'].unique()
```

[32]:
```
array([0, 1, 2])
```

Conclusion- In this way we have explored the functions of the python library for Data Preprocessing, Data Wrangling Techniques and How to Handle missing values on Iris Dataset. In addition to the codes and outputs, explain every operation that you do in the above steps and explain everything that you do to import/read/scrape the data set