Practical 1

Title: Data Wrangling I

[ ]:
```
#import pandas library
import pandas as pd
```

Load dataset

[ ]:
```
#load iris dataset into pandas dataframe
df= pd.read_csv("IRIS.csv")
```

[ ]:
```
#display dataset
df.head()
```

[ ]:

	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

[ ]:
```
#display last rows of dataset
df.tail()
```

[ ]:

	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

[ ]:
```
##dimension of dataframe
df.shape
```

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

[ ]:
```
#dataset info
df.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
```

descriptive statistics

[ ]:
```
df.describe()
```

[ ]:
```
       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
```

Data Preprocessing

Checking for missing values in dataset

[ ]:
```
df.isnull()
```

[ ]:
```
     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]
```

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

[ ]:
```
sepal_length    False
sepal_width     False
petal_length    False
```

```
petal_width     False
species         False
dtype: bool
```

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

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

To check the data type

[ ]:
```
df.dtypes
```

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

Data Formatting

[ ]:
```
#change data type of petal length to int
df['petal_length']=df['petal_length'].astype("int")
```

[ ]:
```
df.dtypes
```

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

Data Normalization using MinMaxScaler

[ ]:
```
#import library
from sklearn import preprocessing
min_max_scaler = preprocessing.MinMaxScaler()
```

[ ]:
```
#separate feature from class label
x=df.iloc[:,:4]
```

[ ]:
```
x
```

[ ]:
```
     sepal_length  sepal_width  petal_length  petal_width
0             5.1          3.5             1          0.2
1             4.9          3.0             1          0.2
2             4.7          3.2             1          0.2
3             4.6          3.1             1          0.2
4             5.0          3.6             1          0.2
..            …            …               …          …
145           6.7          3.0             5          2.3
146           6.3          2.5             5          1.9
147           6.5          3.0             5          2.0
148           6.2          3.4             5          2.3
149           5.9          3.0             5          1.8

[150 rows x 4 columns]
```

[ ]:
```
# create object to transform data to fit minmax processor
x_scaled = min_max_scaler.fit_transform(x)
```

[ ]:
```
#run normalizer on dataframe
df_normalized = pd.DataFrame(x_scaled)
```

[ ]:
```
df_normalized
```

[ ]:
```
            0         1    2         3
0    0.222222  0.625000  0.0  0.041667
1    0.166667  0.416667  0.0  0.041667
2    0.111111  0.500000  0.0  0.041667
3    0.083333  0.458333  0.0  0.041667
4    0.194444  0.666667  0.0  0.041667
..        …       …  …       …
145  0.666667  0.416667  0.8  0.916667
146  0.555556  0.208333  0.8  0.750000
147  0.611111  0.416667  0.8  0.791667
148  0.527778  0.583333  0.8  0.916667
149  0.444444  0.416667  0.8  0.708333

[150 rows x 4 columns]
```

Categorical variable to Quantitative variable using One-Hot Encoding

[ ]:
```
#observe uniques values for species column
new_df=df
new_df['species'].unique()
```

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

[ ]:
```
#import library
from sklearn import preprocessing
```

```
enc = preprocessing.OneHotEncoder()
```

[ ]:
```
#remove target variable from dataset
features_df=new_df.drop(columns=['species'])
```

[ ]:
```
features_df
```

[ ]:
```
     sepal_length  sepal_width  petal_length  petal_width
0             5.1          3.5             1          0.2
1             4.9          3.0             1          0.2
2             4.7          3.2             1          0.2
3             4.6          3.1             1          0.2
4             5.0          3.6             1          0.2
..            …            …               …          …
145           6.7          3.0             5          2.3
146           6.3          2.5             5          1.9
147           6.5          3.0             5          2.0
148           6.2          3.4             5          2.3
149           5.9          3.0             5          1.8

[150 rows x 4 columns]
```

[ ]:
```
#apply one hot encoder for species column
enc_df=(enc. fit_transform(new_df[['species']])).toarray()
enc_df = pd.DataFrame(enc_df, columns =␣
 ↪['Iris-setosa','Iris-versicolor','Iris-virginca'])
```

[ ]:
```
#join encoded values with feature variable
df_encode = features_df.join(enc_df)
```

[ ]:
```
#observe merge dataframe
df_encode
```

[ ]:
```
     sepal_length  sepal_width  petal_length  petal_width  Iris-setosa  \
0             5.1          3.5             1          0.2          1.0
1             4.9          3.0             1          0.2          1.0
2             4.7          3.2             1          0.2          1.0
3             4.6          3.1             1          0.2          1.0
4             5.0          3.6             1          0.2          1.0
..            …            …               …          …            …
145           6.7          3.0             5          2.3          0.0
146           6.3          2.5             5          1.9          0.0
147           6.5          3.0             5          2.0          0.0
148           6.2          3.4             5          2.3          0.0
149           5.9          3.0             5          1.8          0.0

     Iris-versicolor  Iris-virginca
```

```
0                0.0            0.0
1                0.0            0.0
2                0.0            0.0
3                0.0            0.0
4                0.0            0.0
..               …              …
145              0.0            1.0
146              0.0            1.0
147              0.0            1.0
148              0.0            1.0
149              0.0            1.0

[150 rows x 7 columns]
```