Experiment No.7

Title: Data normalization and discretization

Batch:A2 **Roll No.:16010421063** **Experiment No.: 6**

Aim: Data normalization and discretization

Resources needed: Python

Theory:

Normalization: refers to rescaling real-valued numeric attributes into a: 0 to 1 range.
Data normalization: is used in machine learning to make model training less sensitive to the scale of features. This allows our model to converge to better weights and, in turn, leads to a more accurate model. Normalization makes the features more consistent with each other, which allows the model to predict outputs more accurately.

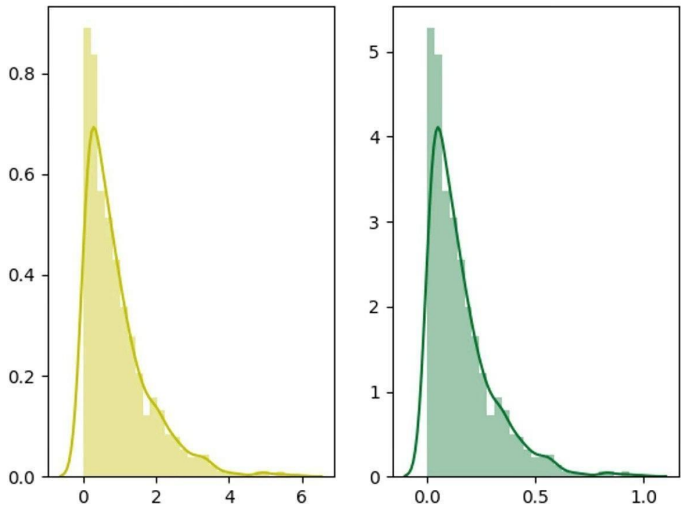


Fig: Left original data Right: Normalized data

Python provides the preprocessing library, which contains the normalize function to normalize the data. It takes an array in as an input and normalizes its values between 0 and 1. It then returns an output array with the same dimensions as the input.

Data Discretization:

Data discretization is the process of converting continuous data into discrete buckets by grouping it. Discretization is also known for easy maintainability of the data. Training a model with discrete data becomes faster and more effective than when attempting the same with continuous data. Although continuous-valued data contains more information, huge amounts of data can slow the model down. Here, discretization can help us strike a balance between both. Some famous methods of data discretization are binning and using a histogram. Although data discretization is useful, we need to effectively pick the range of each bucket, which is a challenge.
The main challenge in discretization is to choose the number of intervals or bins and how to decide on their width. Here we make use of a function called pandas.cut(). This function is useful to achieve the bucketing and sorting of segmented data.

Procedure / Approach /Algorithm / Activity Diagram:

Q1. Data Normalization:

```
from sklearn import preprocessing
import numpy as np
a = np.random.random((1,
4)) a = a*20
print("Data = ", a)
# normalize the data attributes
normalized = preprocessing.normalize(a)
print("Normalized Data = ", normalized)
```

Output: Sample I/P and Normalized O/P
Data = [[0.22135985 16.72816464 19.05510138 2.45065832]]
Normalized Data = [[0.00868925 0.65664645 0.74798789 0.09619801]]

Q2. Activity on Data Discretization of continuous data:

Download Student_bucketing.csv dataset from the url:
https://github.com/TrainingByPackt/Data-Science-with-Python/blob/master/Chapter01/Data / Student_bucketing.csv

Load the Student_bucketing.csv dataset and perform bucketing. The dataset consists of student details such as Student_id, Age, Grade, Employed, and marks. Follow these steps to complete this exercise:

1. Open a Jupyter notebook and add a new cell. Write the following code to import the required libraries and load the dataset into a pandas dataframe:

```
import pandas as pd
dataset=https://github.com/TrainingByPackt/Data-Science-with-Python/blob/master/Chapter01/Data/Student_bucketing.csv"
df = pd.read_csv(dataset, header = 0)
```

2. Once we load the data frame it must display the first five rows of the data frame. Add the following code to do this:

```
df.head()
```

3. Perform bucketing using the pd.cut() function on the marks column and display the top 10 columns. The cut() function takes parameters such as x, bins, and labels. Here, we have used only three parameters. Add the following code to implement this:

```
df['bucket']=pd.cut(df['marks'],5,labels=['Poor','Below_average','Average','Above_Average','Excellent'])
df.head(10)
```

In the preceding code, the first parameter represents an array. Here, we have selected the marks column as an array from the data frame. 5 represents the number of bins to be used. As we have set bins to 5, the labels need to be populated accordingly with five values: Poor, Below_average, Average, Above_average, and Excellent. Here we can see the whole of the continuous marks column is put into five discrete buckets.

Results: Students must submit the output of Q1 and Q2 (in tabular form).

```
from sklearn import preprocessing
import numpy as np
a = np.random.random((1, 4))
a = a*20
print("Data = ", a)
normalized = preprocessing.normalize(a)
print("Normalized Data = ", normalized)
```

```
Data =  [[19.52788477 17.65451309  1.15110101 15.16753296]]
Normalized Data =  [[0.64228151 0.58066542 0.03786027 0.49886745]]
```

+ Code + Markdown | Run All Clear Outputs of All Cells Restart | Variables Outline ···

[1]
```
import pandas as pd
```

```
import pandas as pd
df = pd.read_csv('data.csv', header = 0)
df.head()
```

[6] ✓ 0.3s

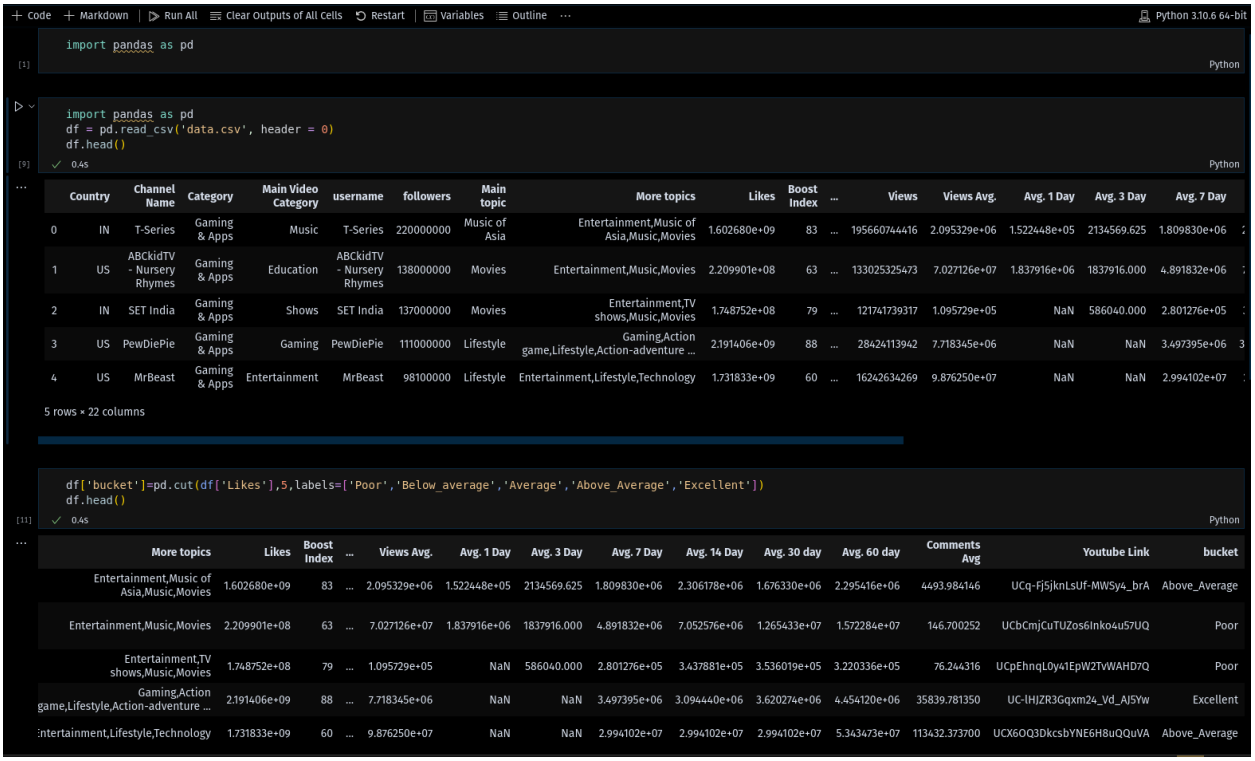
	Student_id	Age	Grade	Employed	marks
0	1	19	1st Class	yes	29
1	2	20	2nd Class	no	41
2	3	18	1st Class	no	57
3	4	21	2nd Class	no	29
4	5	19	1st Class	no	57

```
df['bucket']=pd.cut(df['marks'],5,labels=['Poor','Below_average','Average','Above_Average','Excellent'])
df.head(10)
```

[8] ✓ 0.5s

	Student_id	Age	Grade	Employed	marks	bucket
0	1	19	1st Class	yes	29	Poor
1	2	20	2nd Class	no	41	Below_average
2	3	18	1st Class	no	57	Average
3	4	21	2nd Class	no	29	Poor
4	5	19	1st Class	no	57	Average
5	6	20	2nd Class	yes	53	Average
6	7	19	3rd Class	yes	78	Above_Average
7	8	21	3rd Class	yes	70	Above_Average
8	9	22	3rd Class	yes	97	Excellent
9	10	21	1st Class	no	58	Average

[ ]



```
import pandas as pd
```

```
import pandas as pd
df = pd.read_csv('data.csv', header = 0)
df.head()
```

0.4s

Python

	Country	Channel Name	Category	Main Video Category	username	followers	Main topic	More topics	Likes	Boost Index	...	Views	Views Avg.	Avg. 1 Day	Avg. 3 Day	Avg. 7 Day	
0	IN	T-Series	Gaming & Apps	Music	T-Series	220000000	Music of Asia	Entertainment,Music of Asia,Music,Movies	1.602680e+09	83	...	195660744416	2.095329e+06	1.522448e+05	2134569.625	1.809830e+06	2
1	US	ABCkidTV - Nursery Rhymes	Gaming & Apps	Education	ABCkidTV - Nursery Rhymes	138000000	Movies	Entertainment,Music,Movies	2.209901e+08	63	...	133025325473	7.027126e+07	1.837916e+06	1837916.000	4.891832e+06	1
2	IN	SET India	Gaming & Apps	Shows	SET India	137000000	Movies	Entertainment,TV shows,Music,Movies	1.748752e+08	79	...	121741739317	1.095729e+05	NaN	586040.000	2.801276e+05	1
3	US	PewDiePie	Gaming & Apps	Gaming	PewDiePie	111000000	Lifestyle	Gaming,Action game,Lifestyle,Action-adventure ...	2.191406e+09	88	...	28424113942	7.718345e+06	NaN	NaN	3.497395e+06	3
4	US	MrBeast	Gaming & Apps	Entertainment	MrBeast	98100000	Lifestyle	Entertainment,Lifestyle,Technology	1.731833e+09	60	...	16242634269	9.876250e+07	NaN	NaN	2.994102e+07	1

5 rows × 22 columns

```
df['bucket']=pd.cut(df['Likes'],5,labels=['Poor','Below_average','Average','Above_Average','Excellent'])
df.head()
```

0.4s

Python

	More topics	Likes	Boost Index	...	Views Avg.	Avg. 1 Day	Avg. 3 Day	Avg. 7 Day	Avg. 14 Day	Avg. 30 day	Avg. 60 day	Comments Avg	Youtube Link	bucket
	Entertainment,Music of Asia,Music,Movies	1.602680e+09	83	...	2.095329e+06	1.522448e+05	2134569.625	1.809830e+06	2.306178e+06	1.676330e+06	2.295416e+06	4493.984146	UCq-Fj5jknLsUf-MWSy4_brA	Above_Average
	Entertainment,Music,Movies	2.209901e+08	63	...	7.027126e+07	1.837916e+06	1837916.000	4.891832e+06	7.052576e+06	1.265433e+07	1.572284e+07	146.700252	UCbCmjCuTUZos6Inko4u57UQ	Poor
	Entertainment,TV shows,Music,Movies	1.748752e+08	79	...	1.095729e+05	NaN	586040.000	2.801276e+05	3.437881e+05	3.536019e+05	3.220336e+05	76.244316	UCpEhnqL0y41EpW2TvWAHD7Q	Poor
	Gaming,Action game,Lifestyle,Action-adventure ...	2.191406e+09	88	...	7.718345e+06	NaN	NaN	3.497395e+06	3.094440e+06	3.620274e+06	4.454120e+06	35839.781350	UC-lHJZR3Gqxm24_Vd_AJ5Yw	Excellent
	ntertainment,Lifestyle,Technology	1.731833e+09	60	...	9.876250e+07	NaN	NaN	2.994102e+07	2.994102e+07	2.994102e+07	5.343473e+07	113432.373700	UCX6OQ3DkcsbYNE6H8uQQuVA	Above_Average

Questions:

1. Explain scikit learn library of python and its use.
   Scikit-Learn is a free machine learning library for Python. It supports both supervised and unsupervised machine learning, providing diverse algorithms for classification, regression, clustering, and dimensionality reduction. The library is built using many libraries you may already be familiar with, such as NumPy and SciPy. It also plays well with other libraries, such as Pandas and Seaborn.

2. Explain pandas and Numpy in python.
   pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with 'relationa' or 'labeled' data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python.

   ackage providing fast, flexible, and expressive data structures designed to make working with 'relationa' or 'labeled' data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python.

Outcomes:

Apply transformation required on data to make it suitable for mining

Conclusion: (Conclusion to be based on the objectives and outcomes achieved)

Understood the concept of data normalization and discretisation

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

1. Han, Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann 3nd Edition
2. https://www.educative.io/edpresso/data-normalization-in-python