NORMALIZER, SCALER, BUCKETIZER AND BINARIZER

USING PYSPARK

DR. ALVIN ANG

CONTENTS

I. START A SPARK SESSION

- This article presents common transformations required for data preprocessing and feature engineering.
- These help in preparing data the right way for applying machine learning.

Most of the stuff here are abstracted frrom:

https://www.amazon.com/Learn-PySpark-Python-based-Machine-Learning/dp/1484249607

IPYNB:
https://www.alvinang.sg/s/Normalizer_Scaler_Bucketizer_and_Binarizer_with_PySpark_by_Dr_Alvin_Ang.ipynb

File Used: https://www.alvinang.sg/s/transformations.csv

First, you need to install PySpark into Google Colab.

Follow the steps here:

- https://tatwan.github.io/blog/colab/python/spark/2020/01/06/Colab-Spark-Instructions.html

Or….

```
[4] !apt-get install openjdk-8-jdk-headless -qq > /dev/null

!wget -q https://dlcdn.apache.org/spark/spark-3.2.1/spark-3.2.1-bin-hadoop3.2.tgz

[6] !tar xf spark-3.2.1-bin-hadoop3.2.tgz

[7] !pip install -q findspark

[8] import os
    os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
    os.environ["SPARK_HOME"] = "/content/spark-3.2.1-bin-hadoop3.2"

[9] os.environ["SPARK_HOME"]

'/content/spark-3.2.1-bin-hadoop3.2'

[10] import findspark
     findspark.init()

[11] from pyspark.sql import SparkSession
     spark = SparkSession.builder.master("local[*]").getOrCreate()

[12] print(spark.version)

3.2.1
```

A. IMPORT THE DATA

```
Importing the Transform.csv

[10] from pyspark import SparkFiles

     url = 'https://www.alvinang.sg/s/transformations.csv'
     spark.sparkContext.addFile(url)
     df = spark.read.csv(SparkFiles.get("transformations.csv"), header=True, inferSchema=True)

df.count()

20
```

```
[12] df.show()
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+---------+
|Col 1|Col 2|Col 3|Col 4|Col 5|Col 6|Col 7|Col 8|    label|
+-----+-----+-----+-----+-----+-----+-----+-----+---------+
|    3|   32|  9.0|    3|    3|   17|    2|    5|0.1111111|
|    3|   27| 13.0|    3|    1|   14|    3|    4|3.2307692|
|    4|   22|  2.5|    0|    1|   16|    3|    5|1.3999996|
|    4|   37| 16.5|    4|    3|   16|    5|    5|0.7272727|
|    5|   27|  9.0|    1|    1|   14|    3|    4| 4.666666|
|    4|   27|  9.0|    0|    2|   14|    3|    4| 4.666666|
|    5|   37| 23.0|    6|    2|   12|    5|    4|0.8521735|
|    5|   37| 23.0|    6|    2|   12|    2|    3| 1.826086|
|    3|   22|  2.5|    0|    2|   12|    3|    3|4.7999992|
|    3|   27|  6.0|    0|    1|   16|    3|    5| 1.333333|
|    2|   27|  6.0|    2|    1|   16|    3|    5|3.2666645|
|    5|   27|  6.0|    2|    3|   14|    3|    5| 2.041666|
|    3|   37| 16.5|    6|    1|   12|    2|    3|0.4848484|
|    5|   27|  6.0|    0|    2|   14|    3|    2|      2.0|
|    4|   22|  6.0|    1|    1|   14|    4|    4|3.2666645|
|    4|   37|  9.0|    2|    2|   14|    3|    6|1.3611107|
|    4|   27|  6.0|    1|    1|   12|    3|    5|      2.0|
|    1|   37| 23.0|    6|    4|   14|    5|    2| 1.826086|
|    2|   42| 23.0|    2|    2|   20|    4|    4| 1.826086|
|    4|   37|  6.0|    0|    2|   16|    5|    4| 2.041666|
+-----+-----+-----+-----+-----+-----+-----+-----+---------+
```

Vector Assembler

```
from pyspark.ml.feature import VectorAssembler

assembler = VectorAssembler(inputCols=[col for col in df.columns if col !='label'], \
                            outputCol="features")
```

```
[14] df_new=assembler.transform(df)
```

```
[20] df_new.show()
```

Col 1	Col 2	Col 3	Col 4	Col 5	Col 6	Col 7	Col 8	label	features
3	32	9.0	3	3	17	2	5	0.1111111	[3.0,32.0,9.0,3.0...
3	27	13.0	3	1	14	3	4	3.2307692	[3.0,27.0,13.0,3....
4	22	2.5	0	1	16	3	5	1.3999996	[4.0,22.0,2.5,0.0...
4	37	16.5	4	3	16	5	5	0.7272727	[4.0,37.0,16.5,4....
5	27	9.0	1	1	14	3	4	4.666666	[5.0,27.0,9.0,1.0...
4	27	9.0	0	2	14	3	4	4.666666	[4.0,27.0,9.0,0.0...
5	37	23.0	6	2	12	5	4	0.8521735	[5.0,37.0,23.0,6....
5	37	23.0	6	2	12	2	3	1.826086	[5.0,37.0,23.0,6....
3	22	2.5	0	2	12	3	3	4.7999992	[3.0,22.0,2.5,0.0...
3	27	6.0	0	1	16	3	5	1.333333	[3.0,27.0,6.0,0.0...
2	27	6.0	2	1	16	3	5	3.2666645	[2.0,27.0,6.0,2.0...
5	27	6.0	2	3	14	3	5	2.041666	[5.0,27.0,6.0,2.0...
3	37	16.5	6	1	12	2	3	0.4848484	[3.0,37.0,16.5,6....
5	27	6.0	0	2	14	3	2	2.0	[5.0,27.0,6.0,0.0...
4	22	6.0	1	1	14	4	4	3.2666645	[4.0,22.0,6.0,1.0...
4	37	9.0	2	2	14	3	6	1.3611107	[4.0,37.0,9.0,2.0...
4	27	6.0	1	1	12	3	5	2.0	[4.0,27.0,6.0,1.0...
1	37	23.0	6	4	14	5	2	1.826086	[1.0,37.0,23.0,6....
2	42	23.0	2	2	20	4	4	1.826086	[2.0,42.0,23.0,2....
4	37	6.0	0	2	16	5	4	2.041666	[4.0,37.0,6.0,0.0...

the vector assembler took the first 8 columns and placed it into 'features' column

```
NORMALIZER

[21] from pyspark.ml.feature import Normalizer

[22] normalizer = Normalizer(inputCol="features", outputCol="norm_features", p=1.0)

     normalizer

     Normalizer_fa7d8dd904d0
```

- Normalization refers to transformation of data in such a way that the new normalized data has a mean of 0 and a standard deviation of 1.

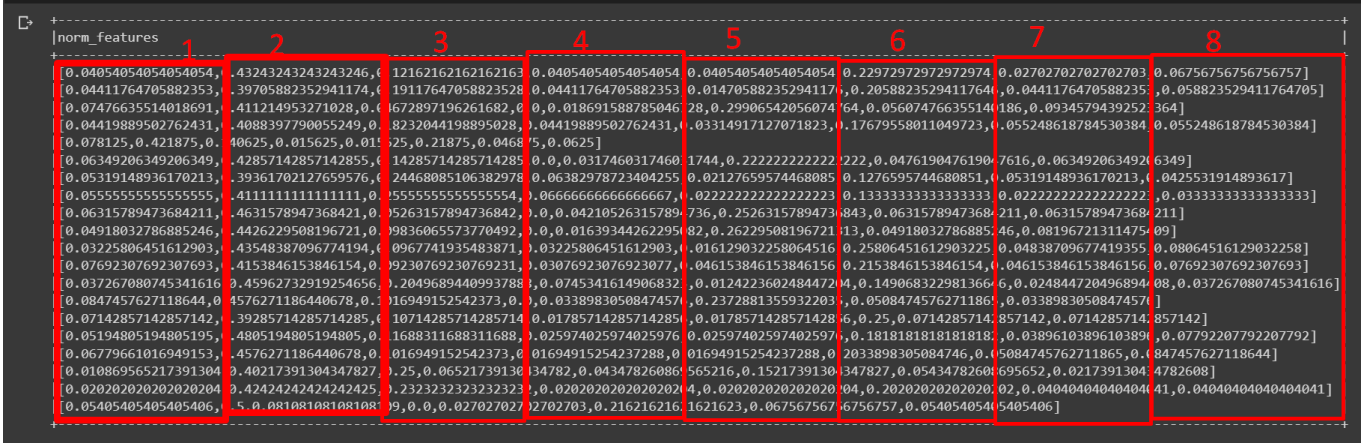
```
[24] normalised_l1_data = normalizer.transform(df_new)

[26] normalised_l1_data.show()
```

Col 1	Col 2	Col 3	Col 4	Col 5	Col 6	Col 7	Col 8	label	features	norm_features
3	32	9.0	3	3	17	2	5	0.1111111	[3.0,32.0,9.0,3.0...	[0.04054054054054...
3	27	13.0	3	1	14	3	4	3.2307692	[3.0,27.0,13.0,3....	[0.04411764705882...
4	22	2.5	0	1	16	3	5	1.3999996	[4.0,22.0,2.5,0.0...	[0.07476635514018...
4	37	16.5	4	3	16	5	5	0.7272727	[4.0,37.0,16.5,4....	[0.04419889502762...
5	27	9.0	1	1	14	3	4	4.666666	[5.0,27.0,9.0,1.0...	[0.078125,0.42187...
4	27	9.0	0	2	14	3	4	4.666666	[4.0,27.0,9.0,0.0...	[0.06349206349206...
5	37	23.0	6	2	12	5	4	0.8521735	[5.0,37.0,23.0,6....	[0.05319148936170...
5	37	23.0	6	2	12	2	3	1.826086	[5.0,37.0,23.0,6....	[0.05555555555555...
3	22	2.5	0	2	12	3	3	4.7999992	[3.0,22.0,2.5,0.0...	[0.06315789473684...
3	27	6.0	0	1	16	3	5	1.333333	[3.0,27.0,6.0,0.0...	[0.04918032786885...
2	27	6.0	2	1	16	3	5	3.2666645	[2.0,27.0,6.0,2.0...	[0.03225806451612...
5	27	6.0	2	3	14	3	5	2.041666	[5.0,27.0,6.0,2.0...	[0.07692307692307...
3	37	16.5	6	1	12	2	3	0.4848484	[3.0,37.0,16.5,6....	[0.03726708074534...
5	27	6.0	0	2	14	3	2	2.0	[5.0,27.0,6.0,0.0...	[0.08474576271186...
4	22	6.0	1	1	14	4	4	3.2666645	[4.0,22.0,6.0,1.0...	[0.07142857142857...
4	37	9.0	2	2	14	3	6	1.3611107	[4.0,37.0,9.0,2.0...	[0.05194805194805...
4	27	6.0	1	1	12	3	5	2.0	[4.0,27.0,6.0,1.0...	[0.0677661016949...
1	37	23.0	6	4	14	5	2	1.826086	[1.0,37.0,23.0,6....	[0.01086956521739...
2	42	23.0	2	2	20	4	4	1.826086	[2.0,42.0,23.0,2....	[0.02020202020202...
4	37	6.0	0	2	16	5	4	2.041666	[4.0,37.0,6.0,0.0...	[0.05405405405405...

norm_features took the 'features' and normalized it....

```
normalised_l1_data.select('norm_features').show(truncate=False)
```



- As can be seen, the 'norm-features' column consists of 8 columns
- These 8 columns show that the previous 8 columns has been "normalized" to fit into values between 0 and 1.

```
from pyspark.ml.feature import Binarizer

binarizer = Binarizer(threshold=0.99, inputCol="label",\
                      outputCol="binarized_label")
```

```
binarizer
```

```
Binarizer_7aaaccf1cb0b
```

```
new_df=binarizer.transform(df)
```

```
new_df.show()
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+----------+---------------+
|Col 1|Col 2|Col 3|Col 4|Col 5|Col 6|Col 7|Col 8|     label|binarized_label|
+-----+-----+-----+-----+-----+-----+-----+-----+----------+---------------+
|    3|   32|  9.0|    3|    3|   17|    2|    5|0.1111111|            0.0|
|    3|   27| 13.0|    3|    1|   14|    3|    4|3.2307692|            1.0|
|    4|   22|  2.5|    0|    1|   16|    3|    5|1.3999996|            1.0|
|    4|   37| 16.5|    4|    3|   16|    5|    5|0.7272727|            0.0|
|    5|   27|  9.0|    1|    1|   14|    3|    4| 4.666666|            1.0|
|    4|   27|  9.0|    0|    2|   14|    3|    4| 4.666666|            1.0|
|    5|   37| 23.0|    6|    2|   12|    5|    4|0.8521735|            0.0|
|    5|   37| 23.0|    6|    2|   12|    2|    3| 1.826086|            1.0|
|    3|   22|  2.5|    0|    2|   12|    3|    3|4.7999992|            1.0|
|    3|   27|  6.0|    0|    1|   16|    3|    5| 1.333333|            1.0|
|    2|   27|  6.0|    2|    1|   16|    3|    5|3.2666645|            1.0|
|    5|   27|  6.0|    2|    3|   14|    3|    5| 2.041666|            1.0|
|    3|   37| 16.5|    6|    1|   12|    2|    3|0.4848484|            0.0|
|    5|   27|  6.0|    0|    2|   14|    3|    2|       2.0|            1.0|
|    4|   22|  6.0|    1|    1|   14|    4|    4|3.2666645|            1.0|
|    4|   37|  9.0|    2|    2|   14|    3|    6|1.3611107|            1.0|
|    4|   27|  6.0|    1|    1|   12|    3|    5|       2.0|            1.0|
|    1|   37| 23.0|    6|    4|   14|    5|    2| 1.826086|            1.0|
|    2|   42| 23.0|    2|    2|   20|    4|    4| 1.826086|            1.0|
|    4|   37|  6.0|    0|    2|   16|    5|    4| 2.041666|            1.0|
```

0 if label starts with 0

1 if label starts with 1 or more

- Because threshold is 0.99
    - Anything below 1 is deemed 0
    - Anything above 1 is deemed 1

V. STANDARD SCALAR

```
from pyspark.ml.feature import StandardScaler

scaler = StandardScaler (inputCol="features",\
                         outputCol="scaled_features",\
                         withStd=False, withMean=True)
```

```
scaler
```

```
StandardScaler_87e40a39e1cc
```

```
scaler_model = scaler.fit(df_new)
```

```
scaler_model
```

```
StandardScalerModel: uid=StandardScaler_87e40a39e1cc, numFeatures=8, withMean=true, withStd=false
```

- Uses the unbiased sample standard deviation instead of the population standard deviation.
- The "unit std" is computed using the corrected sample standard deviation, which is computed as the square root of the unbiased sample variance.
- In other words, after scaling, don't expect it to be between 0 and 1.

```
scaled_data = scaler_model.transform(df_new)
```

```
scaled_data.show()
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+---------+--------------------+--------------------+
|Col 1|Col 2|Col 3|Col 4|Col 5|Col 6|Col 7|Col 8|    label|            features|     scaled_features|
+-----+-----+-----+-----+-----+-----+-----+-----+---------+--------------------+--------------------+
|    3|   32|  9.0|    3|    3|   17|    2|    5|0.1111111|[3.0,32.0,9.0,3.0...|[-0.6500000000000...|
|    3|   27| 13.0|    3|    1|   14|    3|    4|3.2307692|[3.0,27.0,13.0,3....|[-0.6500000000000...|
|    4|   22|  2.5|    0|    1|   16|    3|    5|1.3999996|[4.0,22.0,2.5,0.0...|[0.34999999999999...|
|    4|   37| 16.5|    4|    3|   16|    5|    5|0.7272727|[4.0,37.0,16.5,4....|[0.34999999999999...|
|    5|   27|  9.0|    1|    1|   14|    3|    4| 4.666666|[5.0,27.0,9.0,1.0...|[1.34999999999999...|
|    4|   27|  9.0|    0|    2|   14|    3|    4| 4.666666|[4.0,27.0,9.0,0.0...|[0.34999999999999...|
|    5|   37| 23.0|    6|    2|   12|    5|    4|0.8521735|[5.0,37.0,23.0,6....|[1.34999999999999...|
|    5|   37| 23.0|    6|    2|   12|    2|    3| 1.826086|[5.0,37.0,23.0,6....|[1.34999999999999...|
|    3|   22|  2.5|    0|    2|   12|    3|    3|4.7999992|[3.0,22.0,2.5,0.0...|[-0.6500000000000...|
|    3|   27|  6.0|    0|    1|   16|    3|    5| 1.333333|[3.0,27.0,6.0,0.0...|[-0.6500000000000...|
|    2|   27|  6.0|    2|    1|   16|    3|    5|3.2666645|[2.0,27.0,6.0,2.0...|[-1.6500000000000...|
|    5|   27|  6.0|    2|    3|   14|    3|    5| 2.041666|[5.0,27.0,6.0,2.0...|[1.34999999999999...|
|    3|   37| 16.5|    6|    1|   12|    2|    3|0.4848484|[3.0,37.0,16.5,6....|[-0.6500000000000...|
|    5|   27|  6.0|    0|    2|   14|    3|    2|      2.0|[5.0,27.0,6.0,0.0...|[1.34999999999999...|
|    4|   22|  6.0|    1|    1|   14|    4|    4|3.2666645|[4.0,22.0,6.0,1.0...|[0.34999999999999...|
|    4|   37|  9.0|    2|    2|   14|    3|    6|1.3611107|[4.0,37.0,9.0,2.0...|[0.34999999999999...|
|    4|   27|  6.0|    1|    1|   12|    3|    5|      2.0|[4.0,27.0,6.0,1.0...|[0.34999999999999...|
|    1|   37| 23.0|    6|    4|   14|    5|    2| 1.826086|[1.0,37.0,23.0,6....|[-2.6500000000000...|
|    2|   42| 23.0|    2|    2|   20|    4|    4| 1.826086|[2.0,42.0,23.0,2....|[-1.6500000000000...|
|    4|   37|  6.0|    0|    2|   16|    5|    4| 2.041666|[4.0,37.0,6.0,0.0...|[0.34999999999999...|
```

```
scaled_data.select('scaled_features').show(truncate=False)
```

```
+------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
|scaled_features                                                                                                                                                         |
+------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
|[-0.6500000000000004,1.2500000000000036,-2.049999999999999,0.7500000000000004,1.1500000000000004,2.5500000000000007,-1.35,0.9000000000000004]                           |
|[-0.6500000000000004,-3.7499999999999964,1.9500000000000001,0.7500000000000004,-0.8499999999999996,-0.4499999999999993,-0.3500000000000001,-0.0999999999999964]        |
|[0.3499999999999964,-8.749999999999996,-8.549999999999999,-2.2499999999999996,-0.8499999999999996,1.5500000000000007,-0.3500000000000001,0.9000000000000004]            |
|[0.3499999999999964,6.2500000000000036,5.4500000000000001,1.7500000000000004,1.1500000000000004,1.5500000000000007,1.65,0.9000000000000004]                             |
|[1.3499999999999996,-3.7499999999999964,-2.049999999999999,-1.2499999999999996,-0.8499999999999996,-0.4499999999999993,-0.3500000000000001,-0.0999999999999964]         |
|[0.3499999999999964,-3.7499999999999964,-2.049999999999999,-2.2499999999999996,0.1500000000000036,-0.4499999999999993,-0.3500000000000001,-0.0999999999999964]          |
|[1.3499999999999996,6.2500000000000036,11.950000000000001,3.7500000000000004,0.1500000000000036,-2.4499999999999993,1.65,-0.0999999999999964]                           |
|[1.3499999999999996,6.2500000000000036,11.950000000000001,3.7500000000000004,0.1500000000000036,-2.4499999999999993,-1.35,-1.0999999999999996]                          |
|[-0.6500000000000004,-8.749999999999996,-8.549999999999999,-2.2499999999999996,0.1500000000000036,-2.4499999999999993,-0.3500000000000001,-1.0999999999999996]          |
|[-0.6500000000000004,-3.7499999999999964,-5.049999999999999,-2.2499999999999996,-0.8499999999999996,1.5500000000000007,-0.3500000000000001,0.9000000000000004]          |
|[-1.6500000000000004,-3.7499999999999964,-5.049999999999999,-0.2499999999999956,-0.8499999999999996,1.5500000000000007,-0.3500000000000001,0.9000000000000004]          |
|[1.3499999999999996,-3.7499999999999964,-5.049999999999999,-0.2499999999999956,1.1500000000000004,-0.4499999999999993,-0.3500000000000001,0.9000000000000004]           |
|[-0.6500000000000004,6.2500000000000036,5.4500000000000001,3.7500000000000004,-0.8499999999999996,-2.4499999999999993,-1.35,-1.0999999999999996]                        |
|[1.3499999999999996,-3.7499999999999964,-5.049999999999999,-2.2499999999999996,0.1500000000000036,-0.4499999999999993,-0.3500000000000001,-2.0999999999999996]          |
|[0.3499999999999964,-8.749999999999996,-5.049999999999999,-1.2499999999999996,-0.8499999999999996,-0.4499999999999993,0.6499999999999999,-0.0999999999999964]           |
|[0.3499999999999964,6.2500000000000036,-2.049999999999999,-0.2499999999999956,0.1500000000000036,-0.4499999999999993,-0.3500000000000001,1.9000000000000004]            |
|[0.3499999999999964,-3.7499999999999964,-5.049999999999999,-1.2499999999999996,-0.8499999999999996,-2.4499999999999993,-0.3500000000000001,0.9000000000000004]          |
|[-2.6500000000000004,6.2500000000000036,11.950000000000001,3.7500000000000004,2.1500000000000004,-0.4499999999999993,1.65,-2.0999999999999996]                           |
|[-1.6500000000000004,11.250000000000004,11.950000000000001,-0.2499999999999956,0.1500000000000036,5.5500000000000001,0.6499999999999999,-0.0999999999999964]            |
|[0.3499999999999964,6.2500000000000036,-5.049999999999999,-2.2499999999999996,0.1500000000000036,1.5500000000000007,1.65,-0.0999999999999964]                           |
+------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
```

```
from pyspark.ml.feature import Bucketizer

df.show(10,False)
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+---------+
|Col 1|Col 2|Col 3|Col 4|Col 5|Col 6|Col 7|Col 8|label    |
+-----+-----+-----+-----+-----+-----+-----+-----+---------+
|3    |32   |9.0  |3    |3    |17   |2    |5    |0.1111111|
|3    |27   |13.0 |3    |1    |14   |3    |4    |3.2307692|
|4    |22   |2.5  |0    |1    |16   |3    |5    |1.3999996|
|4    |37   |16.5 |4    |3    |16   |5    |5    |0.7272727|
|5    |27   |9.0  |1    |1    |14   |3    |4    |4.666666 |
|4    |27   |9.0  |0    |2    |14   |3    |4    |4.666666 |
|5    |37   |23.0 |6    |2    |12   |5    |4    |0.8521735|
|5    |37   |23.0 |6    |2    |12   |2    |3    |1.826086 |
|3    |22   |2.5  |0    |2    |12   |3    |3    |4.7999992|
|3    |27   |6.0  |0    |1    |16   |3    |5    |1.333333 |
+-----+-----+-----+-----+-----+-----+-----+-----+---------+
only showing top 10 rows
```

```
splits = [0.0,1.0,2.0,3.0,4.0,5.0,float("inf")]
```

```
splits
```

we are splitting into 5 bins or 5 categories

```
[0.0, 1.0, 2.0, 3.0, 4.0, 5.0, inf]
```

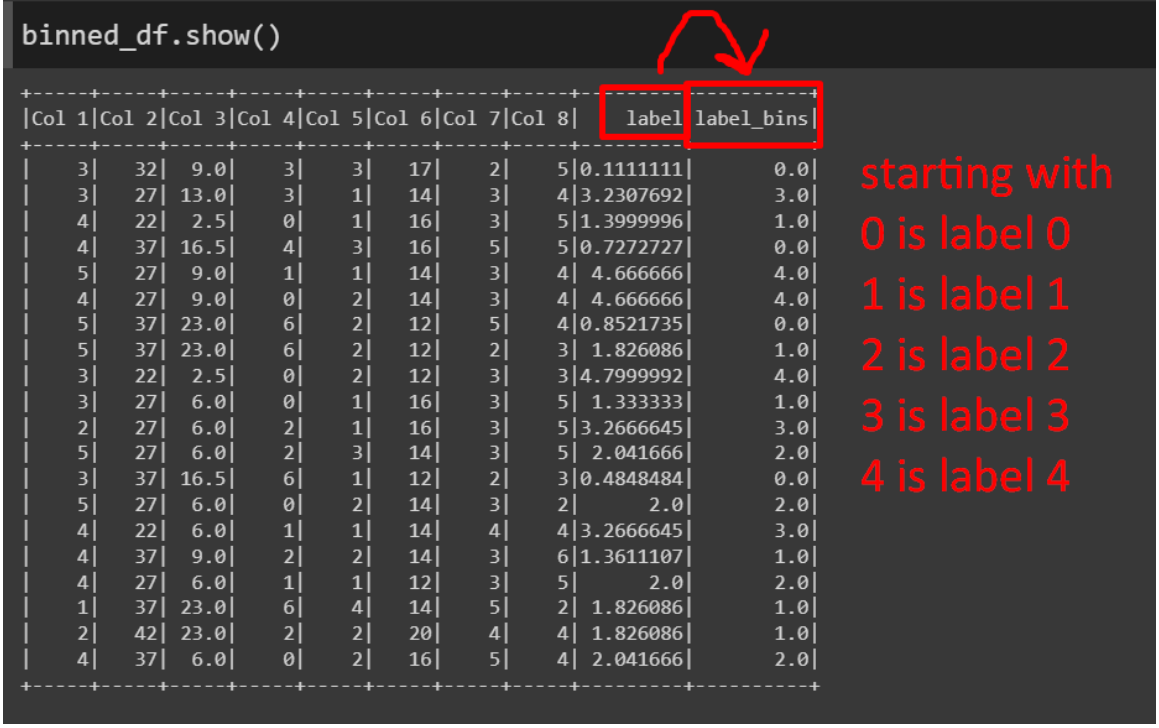
```
bucketizer = Bucketizer(splits=splits, inputCol="label",\
                        outputCol="label_bins")
```

```
bucketizer
```

```
Bucketizer_b788654c444c
```

```
binned_df = bucketizer.transform(df)
```

```
binned_df.show()
```



```
+-----+-----+-----+-----+-----+-----+-----+-----+---------+----------+
|Col 1|Col 2|Col 3|Col 4|Col 5|Col 6|Col 7|Col 8|    label|label_bins|
+-----+-----+-----+-----+-----+-----+-----+-----+---------+----------+
|    3|   32|  9.0|    3|    3|   17|    2|    5|0.1111111|       0.0|
|    3|   27| 13.0|    3|    1|   14|    3|    4|3.2307692|       3.0|
|    4|   22|  2.5|    0|    1|   16|    3|    5|1.3999996|       1.0|
|    4|   37| 16.5|    4|    3|   16|    5|    5|0.7272727|       0.0|
|    5|   27|  9.0|    1|    1|   14|    3|    4| 4.666666|       4.0|
|    4|   27|  9.0|    0|    2|   14|    3|    4| 4.666666|       4.0|
|    5|   37| 23.0|    6|    2|   12|    5|    4|0.8521735|       0.0|
|    5|   37| 23.0|    6|    2|   12|    2|    3| 1.826086|       1.0|
|    3|   22|  2.5|    0|    2|   12|    3|    3|4.7999992|       4.0|
|    3|   27|  6.0|    0|    1|   16|    3|    5| 1.333333|       1.0|
|    2|   27|  6.0|    2|    1|   16|    3|    5|3.2666645|       3.0|
|    5|   27|  6.0|    2|    3|   14|    3|    5| 2.041666|       2.0|
|    3|   37| 16.5|    6|    1|   12|    2|    3|0.4848484|       0.0|
|    5|   27|  6.0|    0|    2|   14|    3|    2|      2.0|       2.0|
|    4|   22|  6.0|    1|    1|   14|    4|    4|3.2666645|       3.0|
|    4|   37|  9.0|    2|    2|   14|    3|    6|1.3611107|       1.0|
|    4|   27|  6.0|    1|    1|   12|    3|    5|      2.0|       2.0|
|    1|   37| 23.0|    6|    4|   14|    5|    2| 1.826086|       1.0|
|    2|   42| 23.0|    2|    2|   20|    4|    4| 1.826086|       1.0|
|    4|   37|  6.0|    0|    2|   16|    5|    4| 2.041666|       2.0|
+-----+-----+-----+-----+-----+-----+-----+-----+---------+----------+
```

```
binned_df.select(['label','label_bins']).show(10,False)
```

```
+---------+----------+
|label    |label_bins|
+---------+----------+
|0.1111111|0.0       |
|3.2307692|3.0       |
|1.3999996|1.0       |
|0.7272727|0.0       |
|4.666666 |4.0       |
|4.666666 |4.0       |
|0.8521735|0.0       |
|1.826086 |1.0       |
|4.7999992|4.0       |
|1.333333 |1.0       |
+---------+----------+
only showing top 10 rows
```

```
binned_df.groupBy('label_bins').count().show()
```

```
+----------+-----+
|label_bins|count|
+----------+-----+
|       0.0|    4|
|       1.0|    6|
|       4.0|    3|
|       3.0|    3|
|       2.0|    4|
+----------+-----+
```

ABOUT DR. ALVIN ANG

Dr. Alvin Ang earned his Ph.D., Masters and Bachelor degrees from NTU, Singapore. He is a scientist, entrepreneur, as well as a personal/business advisor. More about him at www.AlvinAng.sg.