

```
!pip install pyspark
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-w
Collecting pyspark
  Downloading pyspark-3.2.1.tar.gz (281.4 MB)
    |████████████████████████████████████████| 281.4 MB 34 kB/s
Collecting py4j==0.10.9.3
  Downloading py4j-0.10.9.3-py2.py3-none-any.whl (198 kB)
    |████████████████████████████████████████| 198 kB 53.7 MB/s
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.2.1-py2.py3-none-any.whl size=
  Stored in directory: /root/.cache/pip/wheels/9f/f5/07/7cd8017084dce4e93e84e9
Successfully built pyspark
Installing collected packages: py4j, pyspark
Successfully installed py4j-0.10.9.3 pyspark-3.2.1
```

```
from pyspark.sql import SparkSession
spark=SparkSession.builder.appName('SparkML').getOrCreate()
```

```
spark
```

SparkSession - in-memory

SparkContext

[Spark UI](#)

Version

v3.2.1

Master

local[*]

AppName

SparkML

```
df = spark.read.csv('IRIS.csv',header=True,inferSchema=True)
```

```
df.printSchema()
```

```
root
|-- sepal_length: double (nullable = true)
|-- sepal_width: double (nullable = true)
|-- petal_length: double (nullable = true)
|-- petal_width: double (nullable = true)
|-- species: string (nullable = true)
```

```
df.show(5)
```

```
+-----+-----+-----+-----+-----+
|sepal_length|sepal_width|petal_length|petal_width|    species|
+-----+-----+-----+-----+-----+
|         5.1|         3.5|         1.4|         0.2|Iris-setosa|
|         4.9|         3.0|         1.4|         0.2|Iris-setosa|
```

4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5.0	3.6	1.4	0.2	Iris-setosa

only showing top 5 rows

```
#Preprocessing steps
```

```
from pyspark.ml.feature import StringIndexer, OneHotEncoder
```

```
# create object of StringIndexer class and specify input and output column
```

```
SI_species = StringIndexer(inputCol='species',outputCol='species1')
```

```
# transform the data
```

```
df = SI_species.fit(df).transform(df)
```

```
df.tail(5)
```

```
Row(sepal_length=6.7, sepal_width=3.0, petal_length=5.2, petal_width=2.3, spe
Row(sepal_length=6.3, sepal_width=2.5, petal_length=5.0, petal_width=1.9, spe
Row(sepal_length=6.5, sepal_width=3.0, petal_length=5.2, petal_width=2.0, spe
Row(sepal_length=6.2, sepal_width=3.4, petal_length=5.4, petal_width=2.3, spe
Row(sepal_length=5.9, sepal_width=3.0, petal_length=5.1, petal_width=1.8, spe
```

```
df.printSchema()
```

```
root
|-- sepal_length: double (nullable = true)
|-- sepal_width: double (nullable = true)
|-- petal_length: double (nullable = true)
|-- petal_width: double (nullable = true)
|-- species: string (nullable = true)
|-- species1: double (nullable = false)
```

```
from pyspark.ml.feature import VectorAssembler
```

```
fa1=VectorAssembler(inputCols=['sepal_length','sepal_width','petal_length','petal_w
```

```
df1=fa1.transform(df)
```

```
df1.show()
```

sepal_length	sepal_width	petal_length	petal_width	species	species1
5.1	3.5	1.4	0.2	Iris-setosa	0.0
4.9	3.0	1.4	0.2	Iris-setosa	0.0
4.7	3.2	1.3	0.2	Iris-setosa	0.0
4.6	3.1	1.5	0.2	Iris-setosa	0.0
5.0	3.6	1.4	0.2	Iris-setosa	0.0
5.4	3.9	1.7	0.4	Iris-setosa	0.0
4.6	3.4	1.4	0.3	Iris-setosa	0.0
5.0	3.4	1.5	0.2	Iris-setosa	0.0

4.4	2.9	1.4	0.2	Iris-setosa	0.0	[4.4,2
4.9	3.1	1.5	0.1	Iris-setosa	0.0	[4.9,3
5.4	3.7	1.5	0.2	Iris-setosa	0.0	[5.4,3
4.8	3.4	1.6	0.2	Iris-setosa	0.0	[4.8,3
4.8	3.0	1.4	0.1	Iris-setosa	0.0	[4.8,3
4.3	3.0	1.1	0.1	Iris-setosa	0.0	[4.3,3
5.8	4.0	1.2	0.2	Iris-setosa	0.0	[5.8,4
5.7	4.4	1.5	0.4	Iris-setosa	0.0	[5.7,4
5.4	3.9	1.3	0.4	Iris-setosa	0.0	[5.4,3
5.1	3.5	1.4	0.3	Iris-setosa	0.0	[5.1,3
5.7	3.8	1.7	0.3	Iris-setosa	0.0	[5.7,3
5.1	3.8	1.5	0.3	Iris-setosa	0.0	[5.1,3

only showing top 20 rows

```
df2=df1.select("newic1","species1")
```

```
df2.show()
```

newic1	species1
[5.1,3.5,1.4,0.2]	0.0
[4.9,3.0,1.4,0.2]	0.0
[4.7,3.2,1.3,0.2]	0.0
[4.6,3.1,1.5,0.2]	0.0
[5.0,3.6,1.4,0.2]	0.0
[5.4,3.9,1.7,0.4]	0.0
[4.6,3.4,1.4,0.3]	0.0
[5.0,3.4,1.5,0.2]	0.0
[4.4,2.9,1.4,0.2]	0.0
[4.9,3.1,1.5,0.1]	0.0
[5.4,3.7,1.5,0.2]	0.0
[4.8,3.4,1.6,0.2]	0.0
[4.8,3.0,1.4,0.1]	0.0
[4.3,3.0,1.1,0.1]	0.0
[5.8,4.0,1.2,0.2]	0.0
[5.7,4.4,1.5,0.4]	0.0
[5.4,3.9,1.3,0.4]	0.0
[5.1,3.5,1.4,0.3]	0.0
[5.7,3.8,1.7,0.3]	0.0
[5.1,3.8,1.5,0.3]	0.0

only showing top 20 rows

```
df.show(5)
```

sepal_length	sepal_width	petal_length	petal_width	species	species1
5.1	3.5	1.4	0.2	Iris-setosa	0.0
4.9	3.0	1.4	0.2	Iris-setosa	0.0
4.7	3.2	1.3	0.2	Iris-setosa	0.0
4.6	3.1	1.5	0.2	Iris-setosa	0.0
5.0	3.6	1.4	0.2	Iris-setosa	0.0

```
+-----+-----+-----+-----+-----+
only showing top 5 rows
```

```
from pyspark.ml.clustering import KMeans
from pyspark.ml.evaluation import ClusteringEvaluator
train_data,test_data=df2.randomSplit([0.99,0.01])
applyml=KMeans(featuresCol='newic1', k=3)
applyml=applyml.fit(train_data)
```

```
applyml
```

```
KMeansModel: uid=KMeans_babcb527ca9, k=3, distanceMeasure=euclidean, numFeatu
```

```
applyml.transform(train_data).groupBy("prediction").count().show()
```

```
+-----+-----+
|prediction|count|
+-----+-----+
|          1|    62|
|          2|    38|
|          0|    49|
+-----+-----+
```

```
predict=applyml.transform(train_data)
```

```
predict.show()
```

```
↳ +-----+-----+-----+-----+
|          newic1|species1|prediction|
+-----+-----+-----+-----+
|[4.3,3.0,1.1,0.1]|    0.0|         0|
|[4.4,2.9,1.4,0.2]|    0.0|         0|
|[4.4,3.0,1.3,0.2]|    0.0|         0|
|[4.4,3.2,1.3,0.2]|    0.0|         0|
|[4.5,2.3,1.3,0.3]|    0.0|         0|
|[4.6,3.1,1.5,0.2]|    0.0|         0|
|[4.6,3.2,1.4,0.2]|    0.0|         0|
|[4.6,3.4,1.4,0.3]|    0.0|         0|
|[4.6,3.6,1.0,0.2]|    0.0|         0|
|[4.7,3.2,1.3,0.2]|    0.0|         0|
|[4.7,3.2,1.6,0.2]|    0.0|         0|
|[4.8,3.0,1.4,0.1]|    0.0|         0|
|[4.8,3.0,1.4,0.3]|    0.0|         0|
|[4.8,3.1,1.6,0.2]|    0.0|         0|
|[4.8,3.4,1.6,0.2]|    0.0|         0|
|[4.8,3.4,1.9,0.2]|    0.0|         0|
|[4.9,2.4,3.3,1.0]|    1.0|         1|
|[4.9,2.5,4.5,1.7]|    2.0|         1|
|[4.9,3.0,1.4,0.2]|    0.0|         0|
|[4.9,3.1,1.5,0.1]|    0.0|         0|
+-----+-----+-----+-----+
only showing top 20 rows
```

```
predict.groupBy("species1", "prediction").count().show()
```

species1	prediction	count
2.0	1	14
1.0	2	2
2.0	2	36
0.0	0	49
1.0	1	48

```
predict.corr('species1', 'prediction')
```

```
0.9210217528738893
```

✓ 0s completed at 6:22 PM

