Basic

In [1]:
```
import pyspark
from pyspark.sql import SparkSession
```

In [2]:
```
spark = SparkSession.builder.appName('practice').getOrCreate()
```

In [3]:
```
spark
```

Out[3]:

SparkSession - in-memory

SparkContext

Spark UI (http://DESKTOP-AK094UT:4040)

Version

v3.3.1

Master

local[*]

AppName

practice

In [4]:
```
df = spark.read.csv('dummy.csv')
```

In [5]:
```
df.show()
```

```
+----+----+----------+-------+
| _c0| _c1|       _c2|    _c3|
+----+----+----------+-------+
|Name| Age|Experience|Sallary|
| ABC|  22|         5|  30000|
| PQR|  30|         3|  20000|
| XYZ|  24|         6|   null|
| DEF|null|         4|  25000|
| JKL|  20|         3|  15000|
+----+----+----------+-------+
```

In [6]:
```
type(df)
```

Out[6]:
```
pyspark.sql.dataframe.DataFrame
```

In [7]:
```
df1 = spark.read.csv('dummy.csv',inferSchema=True)
```

In [8]:
```
df1.head(4)
```

Out[8]:
```
[Row(_c0='Name', _c1='Age', _c2='Experience', _c3='Sallary'),
 Row(_c0='ABC', _c1='22', _c2='5', _c3='30000'),
 Row(_c0='PQR', _c1='30', _c2='3', _c3='20000'),
 Row(_c0='XYZ', _c1='24', _c2='6', _c3=None)]
```

In [9]:
```
df1.tail(2)
```

Out[9]:
```
[Row(_c0='DEF', _c1=None, _c2='4', _c3='25000'),
 Row(_c0='JKL', _c1='20', _c2='3', _c3='15000')]
```

In [10]:
```
df1.printSchema()
```

```
root
 |-- _c0: string (nullable = true)
 |-- _c1: string (nullable = true)
 |-- _c2: string (nullable = true)
 |-- _c3: string (nullable = true)
```

Datatype

In [11]:
```
df2 = spark.read.csv('dummy.csv',inferSchema=True)
```

In [12]:
```
df2.printSchema()
```

```
root
 |-- _c0: string (nullable = true)
 |-- _c1: string (nullable = true)
 |-- _c2: string (nullable = true)
 |-- _c3: string (nullable = true)
```

First row as column name

In [13]:
```
df3 = spark.read.csv('dummy.csv',header=True,inferSchema=True)
```

In [14]:
```
df3.show()
```

```
+----+----+----------+-------+
|Name| Age|Experience|Sallary|
+----+----+----------+-------+
| ABC|  22|         5|  30000|
| PQR|  30|         3|  20000|
| XYZ|  24|         6|   null|
| DEF|null|         4|  25000|
| JKL|  20|         3|  15000|
+----+----+----------+-------+
```

Column name

In [15]:
```
df3.columns
```

Out[15]:
```
['Name', 'Age', 'Experience', 'Sallary']
```

In [ ]:
```

```

retrive column only

In [16]:
```
df3.select('Name').show()
```

```
+----+
|Name|
+----+
| ABC|
| PQR|
| XYZ|
| DEF|
| JKL|
+----+
```

In [17]:
```
df3.select('Name','Age').show()
```

```
+----+----+
|Name| Age|
+----+----+
| ABC|  22|
| PQR|  30|
| XYZ|  24|
| DEF|null|
| JKL|  20|
+----+----+
```

In [18]:
```
df3.columns
```

Out[18]:
```
['Name', 'Age', 'Experience', 'Sallary']
```

In [19]:
```
df3.dtypes
```

Out[19]:
```
[('Name', 'string'), ('Age', 'int'), ('Experience', 'int'), ('Sallary', 'int')]
```

In [20]:
```
df3.describe().show()
```

```
+-------+----+-----------------+------------------+-----------------+
|summary|Name|              Age|        Experience|          Sallary|
+-------+----+-----------------+------------------+-----------------+
|  count|   5|                4|                 5|                4|
|   mean|null|             24.0|               4.2|          22500.0|
| stddev|null|4.320493798938573|1.3038404810405297|6454.972243679028|
|    min| ABC|               20|                 3|            15000|
|    max| XYZ|               30|                 6|            30000|
+-------+----+-----------------+------------------+-----------------+
```

In [21]:
```
df3.withColumn('Experience after 2 years', df3['Experience']+2).show()
```

```
+----+----+----------+-------+-----------------------+
|Name| Age|Experience|Sallary|Experience after 2 years|
+----+----+----------+-------+-----------------------+
| ABC|  22|         5|  30000|                      7|
| PQR|  30|         3|  20000|                      5|
| XYZ|  24|         6|   null|                      8|
| DEF|null|         4|  25000|                      6|
| JKL|  20|         3|  15000|                      5|
+----+----+----------+-------+-----------------------+
```

Drop column

In [22]:
```
df3 = df3.drop('Experience after 2 years').show()
```

```
+----+----+----------+-------+
|Name| Age|Experience|Sallary|
+----+----+----------+-------+
| ABC|  22|         5|  30000|
| PQR|  30|         3|  20000|
| XYZ|  24|         6|   null|
| DEF|null|         4|  25000|
| JKL|  20|         3|  15000|
+----+----+----------+-------+
```

Column rename

In [24]:
```
df3 = spark.read.csv('dummy.csv',header=True,inferSchema=True)
```

In [25]:
```
df3.withColumnRenamed('Name','New name').show()
```

```
+--------+----+----------+-------+
|New name| Age|Experience|Sallary|
+--------+----+----------+-------+
|     ABC|  22|         5|  30000|
|     PQR|  30|         3|  20000|
|     XYZ|  24|         6|   null|
|     DEF|null|         4|  25000|
|     JKL|  20|         3|  15000|
+--------+----+----------+-------+
```

Drop Row contain null-values

In [26]:
```
df4 = spark.read.csv('dummy.csv',header=True,inferSchema=True)
```

In [27]:
```
df4.show()
```

```
+----+----+----------+-------+
|Name| Age|Experience|Sallary|
+----+----+----------+-------+
| ABC|  22|         5|  30000|
| PQR|  30|         3|  20000|
| XYZ|  24|         6|   null|
| DEF|null|         4|  25000|
| JKL|  20|         3|  15000|
+----+----+----------+-------+
```

In [28]:
```
df4.na.drop().show()
```

```
+----+---+----------+-------+
|Name|Age|Experience|Sallary|
+----+---+----------+-------+
| ABC| 22|         5|  30000|
| PQR| 30|         3|  20000|
| JKL| 20|         3|  15000|
+----+---+----------+-------+
```

Threshold

Delete only those row which having null values more than threshold

In [29]:
```
df6 = spark.read.csv('dummy2.csv',header=True,inferSchema=True)
```

In [31]:
```
df6.show()
```

```
+----+----+----------+-------+
|Name| Age|Experience|Sallary|
+----+----+----------+-------+
| ABC|  22|         5|  30000|
| PQR|  30|         3|  20000|
| XYZ|null|      null|   null|
| DEF|null|         4|  25000|
| JKL|  20|         3|  15000|
+----+----+----------+-------+
```

In [32]:
```
df6.na.drop(how = 'any', thresh=2).show()
```

```
+----+----+----------+-------+
|Name| Age|Experience|Sallary|
+----+----+----------+-------+
| ABC|  22|         5|  30000|
| PQR|  30|         3|  20000|
| DEF|null|         4|  25000|
| JKL|  20|         3|  15000|
+----+----+----------+-------+
```

Threshold on subset of dataframe

In [33]:
```
df6.na.drop(how = 'any', subset=['Age']).show()
```

```
+----+---+----------+-------+
|Name|Age|Experience|Sallary|
+----+---+----------+-------+
| ABC| 22|         5|  30000|
| PQR| 30|         3|  20000|
| JKL| 20|         3|  15000|
+----+---+----------+-------+
```

Filling null values

In [34]:
```
df7 = spark.read.csv('dummy2.csv',header=True,inferSchema=True)
```

In [35]:
```
df7.show()
```

```
+----+----+----------+-------+
|Name| Age|Experience|Sallary|
+----+----+----------+-------+
| ABC|  22|         5|  30000|
| PQR|  30|         3|  20000|
| XYZ|null|      null|   null|
| DEF|null|         4|  25000|
| JKL|  20|         3|  15000|
+----+----+----------+-------+
```

In [43]:
```
# df7.na.fill('Missing').show()
```

In [48]:
```
from pyspark.ml.feature import Imputer

imputer = Imputer(
    inputCols=['Age','Experience','Sallary'],
    outputCols=["{}_imputed".format(c) for c in ['Age','Experience','Sallary']]
).setStrategy("mean")



# imputer = Imputer(
#     inputCols=['Age','Experience','Sallary'],
#     outputCols=["{}_imputed".format(c) for c in ['Age','Experience','Sallary']]
# ).setStrategy("median")

# imputer = Imputer(
#     inputCols=['Age','Experience','Sallary'],
#     outputCols=["{}_imputed".format(c) for c in ['Age','Experience','Sallary']]
# ).setStrategy("mode")
```

In [45]:
```
imputer.fit(df7).transform(df7).show()
```

```
+----+----+----------+-------+-----------+-----------------+---------------+
|Name| Age|Experience|Sallary|Age_imputed|Experience_imputed|Sallary_imputed|
+----+----+----------+-------+-----------+-----------------+---------------+
| ABC|  22|         5|  30000|         22|                5|          30000|
| PQR|  30|         3|  20000|         30|                3|          20000|
| XYZ|null|      null|   null|         24|                3|          22500|
| DEF|null|         4|  25000|         24|                4|          25000|
| JKL|  20|         3|  15000|         20|                3|          15000|
+----+----+----------+-------+-----------+-----------------+---------------+
```

pyspark Filter

In [52]:
```
df8 = spark.read.csv('dummy.csv',header=True,inferSchema=True)
```

In [53]:
```
df8.show()
```

```
+----+---+----------+-------+
|Name|Age|Experience|Sallary|
+----+---+----------+-------+
| ABC| 22|         5|  30000|
| PQR| 30|         3|  20000|
| XYZ| 24|         6|  35000|
| DEF| 26|         4|  25000|
| JKL| 20|         3|  15000|
+----+---+----------+-------+
```

In [54]:
```
df8.filter("Sallary <=25000").show()
```

```
+----+---+----------+-------+
|Name|Age|Experience|Sallary|
+----+---+----------+-------+
| PQR| 30|         3|  20000|
| DEF| 26|         4|  25000|
| JKL| 20|         3|  15000|
+----+---+----------+-------+
```

In [55]:
```
df8.filter("Sallary <=25000").select(['Name','Age']).show()
```

```
+----+---+
|Name|Age|
+----+---+
| PQR| 30|
| DEF| 26|
| JKL| 20|
+----+---+
```

In [59]:
```
df8.filter(df8['Sallary']<=25000).show()
```

```
+----+---+----------+-------+
|Name|Age|Experience|Sallary|
+----+---+----------+-------+
| PQR| 30|         3|  20000|
| DEF| 26|         4|  25000|
| JKL| 20|         3|  15000|
+----+---+----------+-------+
```

In [64]:
```
df8.filter(~(df8['Sallary']<=25000)).show()
```

```
+----+---+----------+-------+
|Name|Age|Experience|Sallary|
+----+---+----------+-------+
| ABC| 22|         5|  30000|
| XYZ| 24|         6|  35000|
+----+---+----------+-------+
```

In [62]:
```
df8.filter((df8['Sallary']<=25000) & (df8['Sallary']>15000)).show()
```

```
+----+---+----------+-------+
|Name|Age|Experience|Sallary|
+----+---+----------+-------+
| PQR| 30|         3|  20000|
| DEF| 26|         4|  25000|
+----+---+----------+-------+
```

In [63]:
```
df8.filter((df8['Sallary']<=25000) | (df8['Sallary']>15000)).show()
```

```
+----+---+----------+-------+
|Name|Age|Experience|Sallary|
+----+---+----------+-------+
| ABC| 22|         5|  30000|
| PQR| 30|         3|  20000|
| XYZ| 24|         6|  35000|
| DEF| 26|         4|  25000|
| JKL| 20|         3|  15000|
+----+---+----------+-------+
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

GroupBy And Aggregate Fun

In [69]:
```
df9 = spark.read.csv('dummy3.csv',header=True,inferSchema=True)
df9.show()
```

```
+----+---+----------+-------+
|Name|Age|Experience|Sallary|
+----+---+----------+-------+
| ABC| 22|         5|  30000|
| PQR| 30|         3|  20000|
| XYZ| 24|         6|  35000|
| DEF| 26|         4|  25000|
| JKL| 20|         3|  15000|
| JKL| 20|         3|   5000|
+----+---+----------+-------+
```

GroupBy

Sum of salary age by similar name

In [71]:
```
df9.groupBy('Name').sum().show()
```

```
+----+--------+---------------+------------+
|Name|sum(Age)|sum(Experience)|sum(Sallary)|
+----+--------+---------------+------------+
| JKL|      40|              6|       20000|
| DEF|      26|              4|       25000|
| PQR|      30|              3|       20000|
| XYZ|      24|              6|       35000|
| ABC|      22|              5|       30000|
+----+--------+---------------+------------+
```

Gropby highets sallary

In [80]:
```
df9.groupBy('Name').mean().show()
```

```
+----+--------+---------------+------------+
|Name|avg(Age)|avg(Experience)|avg(Sallary)|
+----+--------+---------------+------------+
| JKL|    20.0|            3.0|     10000.0|
| DEF|    26.0|            4.0|     25000.0|
| PQR|    30.0|            3.0|     20000.0|
| XYZ|    24.0|            6.0|     35000.0|
| ABC|    22.0|            5.0|     30000.0|
+----+--------+---------------+------------+
```

In [81]:
```
df9.groupBy('Name').count().show()
```

```
+----+-----+
|Name|count|
+----+-----+
| JKL|    2|
| DEF|    1|
| PQR|    1|
| XYZ|    1|
| ABC|    1|
+----+-----+
```

In [87]:
```
df9.groupBy('Name').max().show()
```

```
+----+--------+--------------+------------+
|Name|max(Age)|max(Experience)|max(Sallary)|
+----+--------+--------------+------------+
| JKL|      20|             3|       15000|
| DEF|      26|             4|       25000|
| PQR|      30|             3|       20000|
| XYZ|      24|             6|       35000|
| ABC|      22|             5|       30000|
+----+--------+--------------+------------+
```

In [88]:
```
df9.groupBy('Name').min().show()
```

```
+----+--------+--------------+------------+
|Name|min(Age)|min(Experience)|min(Sallary)|
+----+--------+--------------+------------+
| JKL|      20|             3|        5000|
| DEF|      26|             4|       25000|
| PQR|      30|             3|       20000|
| XYZ|      24|             6|       35000|
| ABC|      22|             5|       30000|
+----+--------+--------------+------------+
```

In [89]:
```
df9.groupBy('Name').avg().show()
```

```
+----+--------+--------------+------------+
|Name|avg(Age)|avg(Experience)|avg(Sallary)|
+----+--------+--------------+------------+
| JKL|    20.0|           3.0|     10000.0|
| DEF|    26.0|           4.0|     25000.0|
| PQR|    30.0|           3.0|     20000.0|
| XYZ|    24.0|           6.0|     35000.0|
| ABC|    22.0|           5.0|     30000.0|
+----+--------+--------------+------------+
```

In [ ]:

In [ ]:

Example of Pyspark ML

In [92]:
```
df10 = spark.read.csv('dummy.csv',header=True,inferSchema=True)
df10.show()
```

```
+----+---+----------+-------+
|Name|Age|Experience|Sallary|
+----+---+----------+-------+
| ABC| 22|         5|  30000|
| PQR| 30|         3|  20000|
| XYZ| 24|         6|  35000|
| DEF| 26|         4|  25000|
| JKL| 20|         3|  15000|
+----+---+----------+-------+
```

In [93]:
```
df10.columns
```

Out[93]:
```
['Name', 'Age', 'Experience', 'Sallary']
```

In [98]:
```
from pyspark.ml.feature import VectorAssembler
featureAssemlber = VectorAssembler(inputCols=['Age','Experience'],outputCol="Independent Fe
```

In [99]:
```
output = featureAssemlber.transform(df10)
```

In [100]:
```
output.show()
```

```
+----+---+----------+-------+-------------------+
|Name|Age|Experience|Sallary|Independent Features|
+----+---+----------+-------+-------------------+
| ABC| 22|         5|  30000|          [22.0,5.0]|
| PQR| 30|         3|  20000|          [30.0,3.0]|
| XYZ| 24|         6|  35000|          [24.0,6.0]|
| DEF| 26|         4|  25000|          [26.0,4.0]|
| JKL| 20|         3|  15000|          [20.0,3.0]|
+----+---+----------+-------+-------------------+
```

In [103]:
```
final_data = output.select("Independent Features","Sallary")
```

In [104]:
```
final_data.show()
```

```
+-------------------+-------+
|Independent Features|Sallary|
+-------------------+-------+
|          [22.0,5.0]|  30000|
|          [30.0,3.0]|  20000|
|          [24.0,6.0]|  35000|
|          [26.0,4.0]|  25000|
|          [20.0,3.0]|  15000|
+-------------------+-------+
```

In [107]:
```
from pyspark.ml.regression import LinearRegression
train_data,test_data = final_data.randomSplit([0.60,0.40])
regressor = LinearRegression(featuresCol='Independent Features', labelCol='Sallary')
regressor= regressor.fit(train_data)
```

In [108]:
```
regressor.coefficients
```

Out[108]:
```
DenseVector([519.4805, 6006.4935])
```

In [110]:
```
regressor.intercept
```

Out[110]:
```
-13262.987012987272
```

In [111]:
```
pred_result = regressor.evaluate(test_data)
```

In [113]:
```
pred_result.predictions.show()
```

```
+--------------------+-------+-----------------+
|Independent Features|Sallary|       prediction|
+--------------------+-------+-----------------+
|          [22.0,5.0]|  30000|28198.05194805193|
+--------------------+-------+-----------------+
```

In [114]:
```
pred_result.meanAbsoluteError
```

Out[114]:
```
1801.9480519480712
```

In [115]:
```
pred_result.meanSquaredError
```

Out[115]:
```
3247016.7819194486
```

In [ ]:
```

```