Pyspark Handling Missing Values

- Dropping Columns
- Dropping Rows
- Various Parameter in Dropping functionalities
- Handling Missing Values by Mean, Median and Mode

In [2]:
```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('Practice').getOrCreate()
```

In [15]:
```
df_pyspark = spark.read.csv('test2.csv', header=True, inferSchema=True)
df_pyspark.show()
```

```
+-------+----+----------+------+
|   name| age|experience|salary|
+-------+----+----------+------+
|Ajinkya|  33|        10| 30000|
| Hitesh|  30|         8| 25000|
|    Jay|  29|         4| 20000|
|  Anish|  24|         3| 20000|
| Nikhil|  21|         1| 15000|
|  Pavan|  23|         2| 18000|
|  Rohit|null|      null| 38000|
|   null|  34|        10| 40000|
|   null|  36|      null|  null|
+-------+----+----------+------+
```

In [6]:
```
## drop the columns
df_pyspark.drop('name').show()
```

```
+----+---------+------+
| age|experince|salary|
+----+---------+------+
|  33|       10| 30000|
|  30|        8| 25000|
|  29|        4| 20000|
|  24|        3| 20000|
|  21|        1| 15000|
|  23|        2| 18000|
|null|     null| 38000|
|  34|       10| 40000|
|  36|     null|  null|
+----+---------+------+
```

In [10]:
```
df_pyspark.show(vertical=True)
```

```
-RECORD 0------------
 name      | Ajinkya
 age       | 33
 experince | 10
 salary    | 30000
-RECORD 1------------
 name      | Hitesh
 age       | 30
 experince | 8
 salary    | 25000
-RECORD 2------------
 name      |  Jay
 age       | 29
 experince | 4
 salary    | 20000
-RECORD 3------------
 name      |  Anish
 age       | 24
 experince | 3
 salary    | 20000
-RECORD 4------------
 name      |  Nikhil
 age       | 21
 experince | 1
 salary    | 15000
-RECORD 5------------
 name      |  Pavan
 age       | 23
 experince | 2
 salary    | 18000
-RECORD 6------------
 name      |  Rohit
 age       | null
 experince | null
 salary    | 38000
-RECORD 7------------
 name      | null
 age       | 34
 experince | 10
 salary    | 40000
-RECORD 8------------
 name      | null
 age       | 36
 experince | null
 salary    | null
```

In [11]:
```
df_pyspark.show()
```

```
+-------+----+---------+------+
|   name| age|experince|salary|
+-------+----+---------+------+
|Ajinkya|  33|       10| 30000|
| Hitesh|  30|        8| 25000|
|    Jay|  29|        4| 20000|
|  Anish|  24|        3| 20000|
| Nikhil|  21|        1| 15000|
|  Pavan|  23|        2| 18000|
|  Rohit|null|     null| 38000|
|   null|  34|       10| 40000|
|   null|  36|     null|  null|
+-------+----+---------+------+
```

In [9]:
```
df_pyspark.na.drop().show()
```

```
+-------+---+---------+------+
|   name|age|experince|salary|
+-------+---+---------+------+
|Ajinkya| 33|       10| 30000|
| Hitesh| 30|        8| 25000|
|    Jay| 29|        4| 20000|
|  Anish| 24|        3| 20000|
| Nikhil| 21|        1| 15000|
|  Pavan| 23|        2| 18000|
+-------+---+---------+------+
```

In [13]:
```
### any == how
df_pyspark.na.drop(how='any').show()
```

```
+-------+---+---------+------+
|   name|age|experince|salary|
+-------+---+---------+------+
|Ajinkya| 33|       10| 30000|
| Hitesh| 30|        8| 25000|
|    Jay| 29|        4| 20000|
|  Anish| 24|        3| 20000|
| Nikhil| 21|        1| 15000|
|  Pavan| 23|        2| 18000|
+-------+---+---------+------+
```

In [14]:
```
### how == all
df_pyspark.na.drop(how='all').show()
```

```
+-------+----+---------+------+
|   name| age|experince|salary|
+-------+----+---------+------+
|Ajinkya|  33|       10| 30000|
| Hitesh|  30|        8| 25000|
|    Jay|  29|        4| 20000|
|  Anish|  24|        3| 20000|
| Nikhil|  21|        1| 15000|
|  Pavan|  23|        2| 18000|
|  Rohit|null|     null| 38000|
|   null|  34|       10| 40000|
|   null|  36|     null|  null|
+-------+----+---------+------+
```

In [21]:
```
### threshold   thresh=2 means at least 2 "NON NULL VALUES" should be present in a row
```

```
df_pyspark.na.drop(how='any', thresh=2).show()
```

```
+-------+----+---------+------+
|   name| age|experince|salary|
+-------+----+---------+------+
|Ajinkya|  33|       10| 30000|
| Hitesh|  30|        8| 25000|
|    Jay|  29|        4| 20000|
|  Anish|  24|        3| 20000|
| Nikhil|  21|        1| 15000|
|  Pavan|  23|        2| 18000|
|  Rohit|null|     null| 38000|
|   null|  34|       10| 40000|
+-------+----+---------+------+
```

In [26]:
```
### subset
### if there are null values in a row of optional list of column names to consider.
### that row will be omitted

df_pyspark.na.drop(how='any', subset=['experience']).show()
```

```
+-------+---+----------+------+
|   name|age|experience|salary|
+-------+---+----------+------+
|Ajinkya| 33|        10| 30000|
| Hitesh| 30|         8| 25000|
|    Jay| 29|         4| 20000|
|  Anish| 24|         3| 20000|
| Nikhil| 21|         1| 15000|
|  Pavan| 23|         2| 18000|
|   null| 34|        10| 40000|
+-------+---+----------+------+
```

In [40]:
```
### Filling the Missing Values
df_pyspark.na.fill('Missing Values').show()
```

```
+--------------+--------------+--------------+--------------+
|          name|           age|    experience|        salary|
+--------------+--------------+--------------+--------------+
|       Ajinkya|            33|            10|         30000|
|        Hitesh|            30|             8|         25000|
|           Jay|            29|             4|         20000|
|         Anish|            24|             3|         20000|
|        Nikhil|            21|             1|         15000|
|         Pavan|            23|             2|         18000|
|         Rohit|Missing Values|Missing Values|         38000|
|Missing Values|            34|            10|         40000|
|Missing Values|            36|Missing Values|Missing Values|
+--------------+--------------+--------------+--------------+
```

In [45]:
```
df_pyspark.na.fill(value='Missing Values', subset=['age', 'experience']).show(truncate=5)
```

```
+-----+-----+----------+------+
| name|  age|experience|salary|
+-----+-----+----------+------+
|Aj...|   33|        10| 30000|
|Hi...|   30|         8| 25000|
|  Jay|   29|         4| 20000|
| A...|   24|         3| 20000|
| N...|   21|         1| 15000|
| P...|   23|         2| 18000|
| R...|Mi...|     Mi...| 38000|
| null|   34|        10| 40000|
| null|   36|     Mi...|  null|
+-----+-----+----------+------+
```

In [4]:
```
df_pyspark.show()
```

```
+-------+----+----------+------+
|   name| age|experience|salary|
+-------+----+----------+------+
|Ajinkya|  33|        10| 30000|
| Hitesh|  30|         8| 25000|
|    Jay|  29|         4| 20000|
|  Anish|  24|         3| 20000|
| Nikhil|  21|         1| 15000|
|  Pavan|  23|         2| 18000|
|  Rohit|null|      null| 38000|
|   null|  34|        10| 40000|
|   null|  36|      null|  null|
+-------+----+----------+------+
```

In [19]:
```
from pyspark.ml.feature import Imputer

imputer = Imputer(
    inputCols=['age', 'experience', 'salary'],
    outputCols=["{}_imputed".format(c) for c in ['age', 'experience', 'salary']]
).setStrategy('mean')
## you can change it to 'Mean', 'Median', 'Mode'
```

In [20]:
```
imputer.fit(df_pyspark).transform(df_pyspark).show()
```

```
+-------+----+----------+------+-----------+-----------------+-------------+
|   name| age|experience|salary|age_imputed|experience_imputed|salary_imputed|
+-------+----+----------+------+-----------+-----------------+-------------+
|Ajinkya|  33|        10| 30000|         33|               10|        30000|
| Hitesh|  30|         8| 25000|         30|                8|        25000|
|    Jay|  29|         4| 20000|         29|                4|        20000|
|  Anish|  24|         3| 20000|         24|                3|        20000|
| Nikhil|  21|         1| 15000|         21|                1|        15000|
|  Pavan|  23|         2| 18000|         23|                2|        18000|
|  Rohit|null|      null| 38000|         28|                5|        38000|
|   null|  34|        10| 40000|         34|               10|        40000|
|   null|  36|      null|  null|         36|                5|        25750|
+-------+----+----------+------+-----------+-----------------+-------------+
```

In [ ]:

In [ ]:

In [ ]: