In [1]:
```
import pyspark
```

getting Spark and initializing it.

In [2]:
```
import findspark as fs
fs.init()
```

In [3]:
```
sc = pyspark.SparkConf()
from  pyspark import SparkContext
from pyspark.sql import SQLContext
```

Starting Spark Session

In [4]:
```
# Starting Pyspark session
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("Walmart").getOrCreate()
```

In [5]:
```
df_spark = spark.read.csv("D:\Pyspark/Walmart.csv",header = True, inferSchema=True) #Data
df_spark.show()
```

```
+-----+----------+------------+------------+-----------+----------+-----------+-----------
-+
|Store|      Date|Weekly_Sales|Holiday_Flag|Temperature|Fuel_Price|        CPI|Unemploymen
t|
+-----+----------+------------+------------+-----------+----------+-----------+-----------
-+
|    1|05-02-2010|   1643690.9|           0|      42.31|     2.572|211.0963582|        8.10
6|
|    1|12-02-2010|  1641957.44|           1|      38.51|     2.548|211.2421698|        8.10
6|
|    1|19-02-2010|  1611968.17|           0|      39.93|     2.514|211.2891429|        8.10
6|
|    1|26-02-2010|  1409727.59|           0|      46.63|     2.561|211.3196429|        8.10
6|
|    1|05-03-2010|  1554806.68|           0|       46.5|     2.625|211.3501429|        8.10
6|
|    1|12-03-2010|  1439541.59|           0|      57.79|     2.667|211.3806429|        8.10
6|
|    1|19-03-2010|  1472515.79|           0|      54.58|      2.72| 211.215635|        8.10
6|
|    1|26-03-2010|  1404429.92|           0|      51.45|     2.732|211.0180424|        8.10
6|
|    1|02-04-2010|  1594968.28|           0|      62.27|     2.719|210.8204499|        7.80
8|
|    1|09-04-2010|  1545418.53|           0|      65.86|      2.77|210.6228574|        7.80
8|
|    1|16-04-2010|  1466058.28|           0|      66.32|     2.808|   210.4887|        7.80
8|
|    1|23-04-2010|  1391256.12|           0|      64.84|     2.795|210.4391228|        7.80
8|
|    1|30-04-2010|  1425100.71|           0|      67.41|      2.78|210.3895456|        7.80
8|
|    1|07-05-2010|  1603955.12|           0|      72.55|     2.835|210.3399684|        7.80
8|
|    1|14-05-2010|   1494251.5|           0|      74.78|     2.854|210.3374261|        7.80
8|
|    1|21-05-2010|  1399662.07|           0|      76.44|     2.826|210.6170934|        7.80
```

```
8|
|    1|28-05-2010|  1432069.95|           0|      80.44|     2.759|210.8967606|       7.80
8|
|    1|04-06-2010|  1615524.71|           0|      80.69|     2.705|211.1764278|       7.80
8|
|    1|11-06-2010|  1542561.09|           0|      80.43|     2.668|211.4560951|       7.80
8|
|    1|18-06-2010|  1503284.06|           0|      84.11|     2.637|211.4537719|       7.80
8|
+-----+----------+------------+------------+-----------+----------+-----------+-----------
-+
only showing top 20 rows
```

In [6]:
```
df_spark.count()  # getting total no. of rows in data
```

Out[6]:
```
6435
```

In [7]:
```
df_spark.describe().show()
```

```
+-------+------------------+----------+------------------+-------------------+------------
------+------------------+-----------------+------------------+
|summary|             Store|      Date|      Weekly_Sales|       Holiday_Flag|       Tempe
rature|        Fuel_Price|              CPI|      Unemployment|
+-------+------------------+----------+------------------+-------------------+------------
------+------------------+-----------------+------------------+
|  count|              6435|      6435|              6435|               6435|
  6435|              6435|             6435|              6435|
|   mean|              23.0|      null|1046964.8775617732|0.06993006993006994| 60.66378243
978229|  3.358606837606832|171.5783938487799| 7.999151048951067|
| stddev|12.988182381175454|      null| 564366.6220536977| 0.2550489443698279|18.444932875
811585|0.45901970719285223|39.35671229566419|1.8758847818627944|
|    min|                 1|01-04-2011|         209986.25|                  0|
-2.06|             2.472|          126.064|             3.879|
|    max|                45|31-12-2010|        3818686.45|                  1|
100.14|             4.468|      227.2328068|            14.313|
+-------+------------------+----------+------------------+-------------------+------------
------+------------------+-----------------+------------------+
```

In [8]:
```
df_spark.printSchema()
```

```
root
 |-- Store: integer (nullable = true)
 |-- Date: string (nullable = true)
 |-- Weekly_Sales: double (nullable = true)
 |-- Holiday_Flag: integer (nullable = true)
 |-- Temperature: double (nullable = true)
 |-- Fuel_Price: double (nullable = true)
 |-- CPI: double (nullable = true)
 |-- Unemployment: double (nullable = true)
```

Highest weekly sale achived in whole data set.

In [9]:
```
df_spark.orderBy(df_spark['Weekly_Sales'].desc()).select(['Date']).head(1)[0]['Date']
```

Out[9]:
```
'24-12-2010'
```

Getting avearge sale of weekly

In [10]:
```
from pyspark.sql.functions import mean
df_spark.select(mean('Weekly_Sales')).show()
```

```
+------------------+
| avg(Weekly_Sales)|
+------------------+
|1046964.8775617732|
+------------------+
```

Getting Max and Min sales

In [11]:
```
from pyspark.sql.functions import min,max
df_spark.select(min('Weekly_Sales')).show()
```

```
+-----------------+
|min(Weekly_Sales)|
+-----------------+
|        209986.25|
+-----------------+
```

In [12]:
```
df_spark.select(max('Weekly_Sales')).show()
```

```
+-----------------+
|max(Weekly_Sales)|
+-----------------+
|       3818686.45|
+-----------------+
```

In [13]:
```
df_spark.show()
```

```
+-----+----------+------------+------------+-----------+----------+-----------+-----------
-+
|Store|      Date|Weekly_Sales|Holiday_Flag|Temperature|Fuel_Price|        CPI|Unemploymen
t|
+-----+----------+------------+------------+-----------+----------+-----------+-----------
-+
|    1|05-02-2010|   1643690.9|           0|      42.31|     2.572|211.0963582|       8.10
6|
|    1|12-02-2010|  1641957.44|           1|      38.51|     2.548|211.2421698|       8.10
6|
|    1|19-02-2010|  1611968.17|           0|      39.93|     2.514|211.2891429|       8.10
6|
|    1|26-02-2010|  1409727.59|           0|      46.63|     2.561|211.3196429|       8.10
6|
|    1|05-03-2010|  1554806.68|           0|       46.5|     2.625|211.3501429|       8.10
6|
|    1|12-03-2010|  1439541.59|           0|      57.79|     2.667|211.3806429|       8.10
6|
|    1|19-03-2010|  1472515.79|           0|      54.58|      2.72| 211.215635|       8.10
6|
|    1|26-03-2010|  1404429.92|           0|      51.45|     2.732|211.0180424|       8.10
6|
|    1|02-04-2010|  1594968.28|           0|      62.27|     2.719|210.8204499|       7.80
8|
|    1|09-04-2010|  1545418.53|           0|      65.86|      2.77|210.6228574|       7.80
8|
|    1|16-04-2010|  1466058.28|           0|      66.32|     2.808|   210.4887|       7.80
8|
|    1|23-04-2010|  1391256.12|           0|      64.84|     2.795|210.4391228|       7.80
```

```
8|
|    1|30-04-2010|  1425100.71|           0|      67.41|      2.78|210.3895456|       7.80
8|
|    1|07-05-2010|  1603955.12|           0|      72.55|     2.835|210.3399684|       7.80
8|
|    1|14-05-2010|   1494251.5|           0|      74.78|     2.854|210.3374261|       7.80
8|
|    1|21-05-2010|  1399662.07|           0|      76.44|     2.826|210.6170934|       7.80
8|
|    1|28-05-2010|  1432069.95|           0|      80.44|     2.759|210.8967606|       7.80
8|
|    1|04-06-2010|  1615524.71|           0|      80.69|     2.705|211.1764278|       7.80
8|
|    1|11-06-2010|  1542561.09|           0|      80.43|     2.668|211.4560951|       7.80
8|
|    1|18-06-2010|  1503284.06|           0|      84.11|     2.637|211.4537719|       7.80
8|
+-----+----------+------------+------------+-----------+----------+-----------+-----------
-+
only showing top 20 rows
```

Spliting Date into Day and Month for analysis.

In [14]:
```
splits_col = pyspark.sql.functions.split(df_spark['Date'],'-')
df = df_spark.select('Weekly_Sales','Holiday_Flag','Temperature','Fuel_Price','CPI','Unemp
                     splits_col.getItem(0).alias('date'),
                     splits_col.getItem(1).alias('month'))
```

In [15]:
```
df.show()
```

```
+------------+------------+-----------+----------+-----------+------------+----+-----+
|Weekly_Sales|Holiday_Flag|Temperature|Fuel_Price|        CPI|Unemployment|date|month|
+------------+------------+-----------+----------+-----------+------------+----+-----+
|   1643690.9|           0|      42.31|     2.572|211.0963582|       8.106|  05|   02|
|  1641957.44|           1|      38.51|     2.548|211.2421698|       8.106|  12|   02|
|  1611968.17|           0|      39.93|     2.514|211.2891429|       8.106|  19|   02|
|  1409727.59|           0|      46.63|     2.561|211.3196429|       8.106|  26|   02|
|  1554806.68|           0|       46.5|     2.625|211.3501429|       8.106|  05|   03|
|  1439541.59|           0|      57.79|     2.667|211.3806429|       8.106|  12|   03|
|  1472515.79|           0|      54.58|      2.72| 211.215635|       8.106|  19|   03|
|  1404429.92|           0|      51.45|     2.732|211.0180424|       8.106|  26|   03|
|  1594968.28|           0|      62.27|     2.719|210.8204499|       7.808|  02|   04|
|  1545418.53|           0|      65.86|      2.77|210.6228574|       7.808|  09|   04|
|  1466058.28|           0|      66.32|     2.808|   210.4887|       7.808|  16|   04|
|  1391256.12|           0|      64.84|     2.795|210.4391228|       7.808|  23|   04|
|  1425100.71|           0|      67.41|      2.78|210.3895456|       7.808|  30|   04|
|  1603955.12|           0|      72.55|     2.835|210.3399684|       7.808|  07|   05|
|   1494251.5|           0|      74.78|     2.854|210.3374261|       7.808|  14|   05|
|  1399662.07|           0|      76.44|     2.826|210.6170934|       7.808|  21|   05|
|  1432069.95|           0|      80.44|     2.759|210.8967606|       7.808|  28|   05|
|  1615524.71|           0|      80.69|     2.705|211.1764278|       7.808|  04|   06|
|  1542561.09|           0|      80.43|     2.668|211.4560951|       7.808|  11|   06|
|  1503284.06|           0|      84.11|     2.637|211.4537719|       7.808|  18|   06|
+------------+------------+-----------+----------+-----------+------------+----+-----+
only showing top 20 rows
```

In [19]:
```
df.printSchema()
```

```
root
 |-- Weekly_Sales: double (nullable = true)
```

```
 |-- Holiday_Flag: integer (nullable = true)
 |-- Temperature: double (nullable = true)
 |-- Fuel_Price: double (nullable = true)
 |-- CPI: double (nullable = true)
 |-- Unemployment: double (nullable = true)
 |-- date: string (nullable = true)
 |-- month: string (nullable = true)
```

Doing StringIndexer on date and month as Vector Assembler is not taking directly.

In [20]:
```
from pyspark.ml.feature import StringIndexer
indexer = StringIndexer(inputCols= ['date','month'],outputCols = ['date_ind','month_ind'])
```

In [21]:
```
df_r = indexer.fit(df).transform(df)
```

In [22]:
```
df_r.show()
```

```
+------------+------------+-----------+----------+-----------+------------+----+-----+----
----+----------+
|Weekly_Sales|Holiday_Flag|Temperature|Fuel_Price|        CPI|Unemployment|date|month|date
_ind|month_ind|
+------------+------------+-----------+----------+-----------+------------+----+-----+----
----+----------+
|   1643690.9|           0|      42.31|     2.572|211.0963582|       8.106|  05|   02|
3.0|       7.0|
|  1641957.44|           1|      38.51|     2.548|211.2421698|       8.106|  12|   02|
8.0|       7.0|
|  1611968.17|           0|      39.93|     2.514|211.2891429|       8.106|  19|   02|
13.0|       7.0|
|  1409727.59|           0|      46.63|     2.561|211.3196429|       8.106|  26|   02|
18.0|       7.0|
|  1554806.68|           0|       46.5|     2.625|211.3501429|       8.106|  05|   03|
3.0|       2.0|
|  1439541.59|           0|      57.79|     2.667|211.3806429|       8.106|  12|   03|
8.0|       2.0|
|  1472515.79|           0|      54.58|      2.72| 211.215635|       8.106|  19|   03|
13.0|       2.0|
|  1404429.92|           0|      51.45|     2.732|211.0180424|       8.106|  26|   03|
18.0|       2.0|
|  1594968.28|           0|      62.27|     2.719|210.8204499|       7.808|  02|   04|
0.0|       0.0|
|  1545418.53|           0|      65.86|      2.77|210.6228574|       7.808|  09|   04|
5.0|       0.0|
|  1466058.28|           0|      66.32|     2.808|   210.4887|       7.808|  16|   04|
10.0|       0.0|
|  1391256.12|           0|      64.84|     2.795|210.4391228|       7.808|  23|   04|
15.0|       0.0|
|  1425100.71|           0|      67.41|      2.78|210.3895456|       7.808|  30|   04|
20.0|       0.0|
|  1603955.12|           0|      72.55|     2.835|210.3399684|       7.808|  07|   05|
22.0|       8.0|
|   1494251.5|           0|      74.78|     2.854|210.3374261|       7.808|  14|   05|
24.0|       8.0|
|  1399662.07|           0|      76.44|     2.826|210.6170934|       7.808|  21|   05|
26.0|       8.0|
|  1432069.95|           0|      80.44|     2.759|210.8967606|       7.808|  28|   05|
28.0|       8.0|
|  1615524.71|           0|      80.69|     2.705|211.1764278|       7.808|  04|   06|
2.0|       3.0|
```

```
|  1542561.09|           0|      80.43|     2.668|211.4560951|       7.808|  11|   06|
7.0|      3.0|
|  1503284.06|           0|      84.11|     2.637|211.4537719|       7.808|  18|   06|
12.0|      3.0|
+------------+------------+-----------+----------+-----------+------------+----+-----+----
----+---------+
only showing top 20 rows
```

VectorAssembler for single column

In [23]:
```
from pyspark.ml.feature import VectorAssembler
feature = VectorAssembler(inputCols = ['Holiday_Flag','Temperature','Fuel_Price','Unemploy
```

In [24]:
```
output = feature.transform(df_r)
```

In [25]:
```
output.show()
```

```
+------------+------------+-----------+----------+-----------+------------+----+-----+----
----+---------+--------------------+
|Weekly_Sales|Holiday_Flag|Temperature|Fuel_Price|        CPI|Unemployment|date|month|date
_ind|month_ind|       Independent_f|
+------------+------------+-----------+----------+-----------+------------+----+-----+----
----+---------+--------------------+
|   1643690.9|           0|      42.31|     2.572|211.0963582|       8.106|  05|   02|
3.0|      7.0|[0.0,42.31,2.572,...|
|  1641957.44|           1|      38.51|     2.548|211.2421698|       8.106|  12|   02|
8.0|      7.0|[1.0,38.51,2.548,...|
|  1611968.17|           0|      39.93|     2.514|211.2891429|       8.106|  19|   02|
13.0|      7.0|[0.0,39.93,2.514,...|
|  1409727.59|           0|      46.63|     2.561|211.3196429|       8.106|  26|   02|
18.0|      7.0|[0.0,46.63,2.561,...|
|  1554806.68|           0|       46.5|     2.625|211.3501429|       8.106|  05|   03|
3.0|      2.0|[0.0,46.5,2.625,8...|
|  1439541.59|           0|      57.79|     2.667|211.3806429|       8.106|  12|   03|
8.0|      2.0|[0.0,57.79,2.667,...|
|  1472515.79|           0|      54.58|      2.72| 211.215635|       8.106|  19|   03|
13.0|      2.0|[0.0,54.58,2.72,8...|
|  1404429.92|           0|      51.45|     2.732|211.0180424|       8.106|  26|   03|
18.0|      2.0|[0.0,51.45,2.732,...|
|  1594968.28|           0|      62.27|     2.719|210.8204499|       7.808|  02|   04|
0.0|      0.0|[0.0,62.27,2.719,...|
|  1545418.53|           0|      65.86|      2.77|210.6228574|       7.808|  09|   04|
5.0|      0.0|[0.0,65.86,2.77,7...|
|  1466058.28|           0|      66.32|     2.808|   210.4887|       7.808|  16|   04|
10.0|      0.0|[0.0,66.32,2.808,...|
|  1391256.12|           0|      64.84|     2.795|210.4391228|       7.808|  23|   04|
15.0|      0.0|[0.0,64.84,2.795,...|
|  1425100.71|           0|      67.41|      2.78|210.3895456|       7.808|  30|   04|
20.0|      0.0|[0.0,67.41,2.78,7...|
|  1603955.12|           0|      72.55|     2.835|210.3399684|       7.808|  07|   05|
22.0|      8.0|[0.0,72.55,2.835,...|
|   1494251.5|           0|      74.78|     2.854|210.3374261|       7.808|  14|   05|
24.0|      8.0|[0.0,74.78,2.854,...|
|  1399662.07|           0|      76.44|     2.826|210.6170934|       7.808|  21|   05|
26.0|      8.0|[0.0,76.44,2.826,...|
|  1432069.95|           0|      80.44|     2.759|210.8967606|       7.808|  28|   05|
28.0|      8.0|[0.0,80.44,2.759,...|
|  1615524.71|           0|      80.69|     2.705|211.1764278|       7.808|  04|   06|
2.0|      3.0|[0.0,80.69,2.705,...|
|  1542561.09|           0|      80.43|     2.668|211.4560951|       7.808|  11|   06|
7.0|      3.0|[0.0,80.43,2.668,...|
```

```
|  1503284.06|           0|      84.11|     2.637|211.4537719|       7.808|  18|   06|
12.0|      3.0|[0.0,84.11,2.637,...|
+------------+------------+-----------+----------+-----------+------------+----+-----+----
----+---------+--------------------+
only showing top 20 rows
```

In [90]:
```
final_data = output.select('Independent_f','Weekly_Sales') #required columns for further p
```

In [91]:
```
from pyspark.ml.regression import LinearRegression
train_data,test_data = final_data.randomSplit([0.75,0.25]) #splited data in ration for tra
regression = LinearRegression(featuresCol='Independent_f',labelCol='Weekly_Sales')
regressor = regression.fit(train_data)
```

In [92]:
```
pred_result = regressor.evaluate(test_data)
pred_result.predictions.show()
```

```
+--------------------+------------+------------------+
|       Independent_f|Weekly_Sales|        prediction|
+--------------------+------------+------------------+
|[0.0,10.11,3.008,...|   513372.17| 979846.3561260055|
|[0.0,10.91,3.243,...|  1083657.61|1221453.2041986482|
|[0.0,11.17,3.24,7...|   911807.02|1159403.7331957316|
|[0.0,11.17,3.331,...|   653845.45|1041774.1357079161|
|[0.0,11.29,2.974,...|   816603.05|1210371.4961085916|
|[0.0,11.32,2.911,...|    547384.9|  990335.964986274|
|[0.0,12.19,3.173,...|  1059715.27|1014352.6737697318|
|[0.0,12.98,3.232,...|   809833.21|1111182.8527401455|
|[0.0,13.29,3.24,5...|   904261.65|1273891.7431214233|
|[0.0,14.44,3.331,...|   574798.86|1134832.9246236186|
|[0.0,14.56,3.011,...|   1179420.5|1096655.4517124088|
|[0.0,15.33,3.542,...|  1146992.13|1287040.9964644588|
|[0.0,15.58,3.232,...|  1110706.06|1225533.0353949834|
|[0.0,16.7,3.215,7...|   812323.29|1115530.9121065892|
|[0.0,16.94,2.891,...|  1744544.39|1147290.4568704616|
|[0.0,17.46,3.101,...|   546690.84| 990761.7756411426|
|[0.0,18.49,3.437,...|   977070.62|1245549.8346721989|
|[0.0,20.67,3.437,...|  1301185.28| 1244923.869150405|
|[0.0,20.7,3.372,8...|   558963.83| 990108.0083768124|
|[0.0,20.79,3.24,9...|  1055841.24| 1106201.624576393|
+--------------------+------------+------------------+
only showing top 20 rows
```

In [ ]:
```
```