Linear Reg

Linear Regression Part - Used for Graph 5 on dashboard

Creating the Linear Regression dataset for finding the relation in the sales amount for last 7 days to see if there is a trend

Reading the day level transaction data present on S3

In [ ]:
```
date_reg = sqlContext.read.format('com.databricks.spark.csv') \
    .option("inferSchema",True).option("header",True).load('s3://bigdataprjct/historical_data/date_lvl/date_lvl.csv')
```

Checking the read data

Checking the read data

In [148]:
```
date_reg.limit(5).toPandas()
```

Out[148]:

	date	dist_str	dist_item	sales	cntItmSoldOnPromo	dcoilwtico	holidayFl	cnt_trns
0	8/1/2015	51	3023	1044894.790	1448	45.25	0	105307
1	8/2/2015	51	2982	1043495.475	2194	45.25	0	97134
2	8/3/2015	51	3020	811119.834	1405	45.25	0	89639
3	8/4/2015	51	3014	726613.358	3489	45.75	0	85613

	date	dist_str	dist_item	sales	cntItmSoldOnPromo	dcoilwtico	holidayFl	cnt_trns
4	8/5/2015	51	3019	724346.594	8919	45.13	1	84912

Reading the ml feature Vector Assembler for later use in regression if necessary

In [149]:
```
from pyspark.ml.feature import VectorAssembler
```

Adding a numeric sequence column to the dataset to later filter based on this column to create separate dataframe for capturing relation based on lag days sales.

In [160]:
```
from pyspark.sql.functions import monotonically_increasing_id

date_reg = date_reg.withColumn('Seq',monotonically_increasing_id())
```

Registering a temp table from above daat for further processing

In [180]:
```
date_reg.registerTempTable("date1")
```

Creating sepearate dataframes based on sequence number that are corresponding to different dates

In [269]:
```
dt1 = sqlContext.sql("""
        SELECT sales as sale1 from
        date1
        where Seq <= 742 AND Seq >= 563
        """)

dt2 = sqlContext.sql("""
        SELECT sales as sale2 from
        date1
        where Seq <= 741 AND Seq >= 562
        """)
dt3 = sqlContext.sql("""
        SELECT sales as sale3 from
```

```
        date1
        where Seq <= 740 AND Seq >= 561
        """)

dt4 = sqlContext.sql("""
        SELECT sales as sale4 from
        date1
        where Seq <= 739 AND Seq >= 560
        """)

dt5 = sqlContext.sql("""
        SELECT sales as sale5 from
        date1
        where Seq <= 738 AND Seq >= 559
        """)

dt6 = sqlContext.sql("""
        SELECT sales as sale6 from
        date1
        where Seq <= 737 AND Seq >= 558
        """)

dt7 = sqlContext.sql("""
        SELECT sales as sale7 from
        date1
        where Seq <= 736 AND Seq >= 557
        """)

dt8 = sqlContext.sql("""
        SELECT sales as sale8 from
        date1
        where Seq <= 735 AND Seq >= 556
        """)


dt1.show()
```

```
+-----------+
|      sale1|
+-----------+
```

```
| 645968.257|
| 758855.713|
|1063394.844|
| 858897.123|
|  759120.06|
| 717806.677|
| 739388.035|
|   675578.5|
| 816197.499|
|1026011.223|
| 775708.988|
| 744956.948|
| 949366.829|
| 1008521.71|
| 836225.179|
| 882639.775|
|1125736.347|
| 1196983.69|
| 790143.241|
|  761866.73|
+-----------+
only showing top 20 rows
```

Adding sequence number again to these separated datasets so that they can be joined together with the lag to see if there is a corelation

In [270]:
```
dt1 = dt1.withColumn('Tseq',monotonically_increasing_id())
dt2 = dt2.withColumn('Tseq',monotonically_increasing_id())
dt3 = dt3.withColumn('Tseq',monotonically_increasing_id())
dt4 = dt4.withColumn('Tseq',monotonically_increasing_id())
dt5 = dt5.withColumn('Tseq',monotonically_increasing_id())
dt6 = dt6.withColumn('Tseq',monotonically_increasing_id())
dt7 = dt7.withColumn('Tseq',monotonically_increasing_id())
dt8 = dt8.withColumn('Tseq',monotonically_increasing_id())


dt1.show()
```

```
+-----------+----+
```

```
|      sale1|Tseq|
+-----------+----+
| 645968.257|   0|
| 758855.713|   1|
|1063394.844|   2|
| 858897.123|   3|
|  759120.06|   4|
| 717806.677|   5|
| 739388.035|   6|
|   675578.5|   7|
| 816197.499|   8|
|1026011.223|   9|
| 775708.988|  10|
| 744956.948|  11|
| 949366.829|  12|
| 1008521.71|  13|
| 836225.179|  14|
| 882639.775|  15|
|1125736.347|  16|
| 1196983.69|  17|
| 790143.241|  18|
|  761866.73|  19|
+-----------+----+
only showing top 20 rows
```

Registering the above dataframes as tables for joining them together

In [271]:
```
dt1.registerTempTable("dt1")
dt2.registerTempTable("dt2")
dt3.registerTempTable("dt3")
dt4.registerTempTable("dt4")
dt5.registerTempTable("dt5")
dt6.registerTempTable("dt6")
dt7.registerTempTable("dt7")
dt8.registerTempTable("dt8")
```

Creating the dataset by joining the above different dates dataset

In [272]:
```
dt11 = sqlContext.sql("""
                SELECT dt1.Tseq, dt1.sale1, dt2.sale2
                FROM dt1 INNER JOIN dt2
                ON dt1.Tseq = dt2.Tseq
                """)

dt11.registerTempTable("dt11")

dt12 = sqlContext.sql("""
                SELECT dt11.*, dt3.sale3
                FROM dt11 INNER JOIN dt3
                ON dt11.Tseq = dt3.Tseq
                """)

dt12.registerTempTable("dt12")

dt13 = sqlContext.sql("""
                SELECT dt12.*, dt4.sale4
                FROM dt12 INNER JOIN dt4
                ON dt12.Tseq = dt4.Tseq
                """)

dt13.registerTempTable("dt13")

dt14 = sqlContext.sql("""
                SELECT dt13.*, dt5.sale5
                FROM dt13 INNER JOIN dt5
                ON dt13.Tseq = dt5.Tseq
                """)

dt14.registerTempTable("dt14")

dt15 = sqlContext.sql("""
                SELECT dt14.*, dt6.sale6
                FROM dt14 INNER JOIN dt6
                ON dt14.Tseq = dt6.Tseq
                """)
```

```
dt15.registerTempTable("dt15")

dt16 = sqlContext.sql("""
                SELECT dt15.*, dt7.sale7
                FROM dt15 INNER JOIN dt7
                ON dt15.Tseq = dt7.Tseq
                """)

dt16.registerTempTable("dt16")

dt17 = sqlContext.sql("""
                SELECT dt16.*, dt8.sale8
                FROM dt16 INNER JOIN dt8
                ON dt16.Tseq = dt8.Tseq
                """)

dt17.registerTempTable("dt17")
dt17.show()
```

```
+----+-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+
|Tseq|      sale1|      sale2|      sale3|      sale4|      sale5|      sale6|      sale7|      sale8|
+----+-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+
|   0| 645968.257| 745780.355| 642613.187| 794314.432|  1074820.9| 944924.653|  752207.24| 624387.488|
|   1| 758855.713| 645968.257| 745780.355| 642613.187| 794314.432|  1074820.9| 944924.653|  752207.24|
|   2|1063394.844| 758855.713| 645968.257| 745780.355| 642613.187| 794314.432|  1074820.9| 944924.653|
|   3| 858897.123|1063394.844| 758855.713| 645968.257| 745780.355| 642613.187| 794314.432|  1074820.9|
|   4|  759120.06| 858897.123|1063394.844| 758855.713| 645968.257| 745780.355| 642613.187| 794314.432|
|   5| 717806.677|  759120.06| 858897.123|1063394.844| 758855.713| 645968.257| 745780.355| 642613.187|
|   6| 739388.035| 717806.677|  759120.06| 858897.123|1063394.844| 758855.713| 645968.257| 745780.355|
|   7|   675578.5| 739388.035| 717806.677|  759120.06| 858897.123|1063394.844| 758855.713| 645968.257|
|   8| 816197.499|   675578.5| 739388.035| 717806.677|  759120.06| 858897.123|1063394.844| 758855.713|
|   9|1026011.223| 816197.499|   675578.5| 739388.035| 717806.677|  759120.06| 858897.123|1063394.844|
|  10| 775708.988|1026011.223| 816197.499|   675578.5| 739388.035| 717806.677|  759120.06| 858897.123|
|  11| 744956.948| 775708.988|1026011.223| 816197.499|   675578.5| 739388.035| 717806.677|  759120.06|
|  12| 949366.829| 744956.948| 775708.988|1026011.223| 816197.499|   675578.5| 739388.035| 717806.677|
|  13| 1008521.71| 949366.829| 744956.948| 775708.988|1026011.223| 816197.499|   675578.5| 739388.035|
|  14| 836225.179| 1008521.71| 949366.829| 744956.948| 775708.988|1026011.223| 816197.499|   675578.5|
|  15| 882639.775| 836225.179| 1008521.71| 949366.829| 744956.948| 775708.988|1026011.223| 816197.499|
|  16|1125736.347| 882639.775| 836225.179| 1008521.71| 949366.829| 744956.948| 775708.988|1026011.223|
|  17| 1196983.69|1125736.347| 882639.775| 836225.179| 1008521.71| 949366.829| 744956.948| 775708.988|
```

```
|  18| 790143.241| 1196983.69|1125736.347| 882639.775| 836225.179| 1008521.71| 949366.829| 744956.948|
|  19|  761866.73| 790143.241| 1196983.69|1125736.347| 882639.775| 836225.179| 1008521.71| 949366.829|
+----+-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+
only showing top 20 rows
```

Importing the various libraries required for running a regression

In [273]:
```
import pyspark.mllib
import pyspark.mllib.regression
from pyspark.mllib.regression import LabeledPoint
from pyspark.sql.functions import *
```

Selecting the variables needed for running the model

In [274]:
```
dt17 = dt17.select("sale1","sale2","sale3","sale4","sale5","sale6","sale7","sale8")
```

Converting the dataframe into the RDD

In [296]:
```
dt18 = dt17.rdd
```

Running the model on teh above created dataframe

In [297]:
```
from pyspark.mllib.regression import LinearRegressionWithSGD
dt18 = dt18.map(lambda line:LabeledPoint(line[0],[line[1:]]))
linearModel = LinearRegressionWithSGD.train(dt18,10,.2)
linearModel.weights
```

Out[297]:
```
DenseVector([-1.0466, -1.0459, -1.0461, -1.0479, -1.0496, -1.0495, -1.0482])
```