

DA228- BigData Systems Assignment-5

Importing the packages

```
1 from pyspark.sql.functions import sha2, concat
2 from pyspark.sql.functions import col,lit
3 from pyspark.sql.types import StructType, StructField, StringType,DateType,DecimalType,IntegerType,ArrayType,LongType,BooleanType,DoubleType,FloatType
4 from pyspark.sql.functions import to_date, to_timestamp
5 from datetime import datetime,date,timedelta
6 from pyspark.sql.functions import col
7 from pyspark.sql import functions as F
8 from pyspark.sql import *
9 from pytz import timezone, utc
```

```
from pyspark.sql.functions import sha2, concat
from pyspark.sql.functions import col,lit
from pyspark.sql.types import StructType, StructField,
StringType,DateType,DecimalType,IntegerType,ArrayType,LongType,BooleanType,DoubleType,Float
Type
from pyspark.sql.functions import to_date, to_timestamp
from datetime import datetime,date,timedelta
from pyspark.sql.functions import col
from pyspark.sql import functions as F
from pyspark.sql import *
from pytz import timezone, utc
```

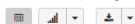
Initializing the Data types

```
1 #Initializing the datatypes
2 schema_response = StructType([
3     StructField("Rest_Id",IntegerType(),True),
4     StructField("Rest_Name",StringType(),True),
5     StructField("Address",StringType(),True),
6     StructField("City",StringType(),True),
7     StructField("State",StringType(),True),
8     StructField("Zipcode",StringType(),True),
9     StructField("Latitude",FloatType(),True),
10    StructField("Longitude",FloatType(),True),
11    StructField("Unknown_values",StringType(),True),
12    StructField("Inspection_ID",StringType(),True),
13    StructField("Inspection_date",StringType(),True),
14    StructField("Inspection_Time",StringType(),True),
15    StructField("Inspection_Points",IntegerType(),True),
16    StructField("Inspection_schedule",StringType(),True),
17    StructField("Response_ID",StringType(),True),
18    StructField("Response_factor",StringType(),True),
19    StructField("Risk_factor",StringType(),True)
20 ])
21
22 #Merging the Restaurant Datasets of 2016,2017 and 2018
23 df=spark.read.format("csv").option("Header","False").option("sep","\\t").schema(schema_response).load("dbfs:/FileStore/shared_uploads/harshitharamesh97@gmail.com/8085/*.txt")
24
25
26 display(df)
27
28 df.createOrReplaceTempView("ReviewDataSet")
```

▶ (1) Spark Jobs

	Rest_Id	Rest_Name	Address	City	State	Zipcode	Latitude	Longitude
1	2764	AL-HAMRA	3083 16th St	San Francisco	CA	94103	37.76491	-122.42135
2	1154	SUNFLOWER RESTAURANT	506 Valencia St	San Francisco	CA	94103	37.76468	-122.421906
3	69665	Shalimar Restaurant	532 Jones St	San Francisco	CA	94102	null	null
4	1154	SUNFLOWER RESTAURANT	506 Valencia St	San Francisco	CA	94103	37.76468	-122.421906
5	1154	SUNFLOWER RESTAURANT	506 Valencia St	San Francisco	CA	94103	37.76468	-122.421906
6	2749	TAQUERIA CANCUN	3211 MISSION St	San Francisco	CA	94110	37.745434	-122.419945
	2638	SF BAGEL CO. (KATZ BAGELS)	3147 16th St	San Francisco	CA	94103	37.764812	-122.42309

Truncated results, showing first 1000 rows.



DA228- BigData Systems
Assignment-5

#Initializing the datatypes

```
schema_response = StructType([
    StructField("Rest_Id", IntegerType(), True),
    StructField("Rest_Name", StringType(), True),
    StructField("Address", StringType(), True),
    StructField("City", StringType(), True),
    StructField("State", StringType(), True),
    StructField("Zipcode", StringType(), True),
    StructField("Latitude", FloatType(), True),
    StructField("Longitude", FloatType(), True),
    StructField("Unknown_values", StringType(), True),
    StructField("Inspection_ID", StringType(), True),
    StructField("Inspection_date", StringType(), True),
    StructField("Inspection_Time", StringType(), True),
    StructField("Inspection_Points", IntegerType(), True),
    StructField("Inspection_schedule", StringType(), True),
    StructField("Response_ID", StringType(), True),
    StructField("Response_factor", StringType(), True),
    StructField("Risk_factor", StringType(), True)
])
```

#Merging the Resturant Datasets of 2016,2017 and 2018

```
df=spark.read.format("csv").option("Header","False").option("sep","\\t").schema(schema_response).load(
    "dbfs:/FileStore/shared_uploads/akshaya6597@gmail.com/BDBA/*.txt")
```

```
display(df)
```

```
df.createOrReplaceTempView("ReviewDataSet")
```

Dropping Unknown values since it's junk

```
df = df.drop(df.Unknown_values)
display(df)
```

Cmd 3

```
1 df = df.drop(df.Unknown_values)
2 display(df)
```

DA228- BigData Systems Assignment-5

► (1) Spark Jobs

	Rest_Id ▲	Rest_Name ▲	Address ▲	City ▲	State ▲	Zipcode ▲	Latitude ▲	Longitude ▲
1	2764	AL-HAMRA	3083 16th St	San Francisco	CA	94103	37.76491	-122.42135
2	1154	SUNFLOWER RESTAURANT	506 Valencia St	San Francisco	CA	94103	37.76468	-122.421906
3	69665	Shalimar Restaurant	532 Jones St	San Francisco	CA	94102	null	null
4	1154	SUNFLOWER RESTAURANT	506 Valencia St	San Francisco	CA	94103	37.76468	-122.421906
5	1154	SUNFLOWER RESTAURANT	506 Valencia St	San Francisco	CA	94103	37.76468	-122.421906
6	2749	TAQUERIA CANCUN	3211 MISSION St	San Francisco	CA	94110	37.745434	-122.419945
7	2638	SF BAGEL CO. (KATZ BAGELS)	3147 16th St	San Francisco	CA	94103	37.764812	-122.42309

Truncated results, showing first 1000 rows.

Creating a View for the all the merged dataset

%sql

CREATE OR REPLACE TEMPORARY VIEW ReviewDataSet_Format AS

```
(
  SELECT
    Rest_Id,
    Rest_Name,
    Address,
    City,
    State,
    Zipcode,
    Latitude,
    Longitude,
    Inspection_ID,
    TO_DATE(RIGHT(Inspection_date, 8), "yyyymmdd") AS Inspection_date,
    TO_TIMESTAMP(Inspection_Time) AS Inspection_Time,
    Inspection_Points,
    Inspection_schedule,
    RIGHT(Response_ID,CHARINDEX('_', (REVERSE(Response_ID)))) - 1) AS Response_ID,
    Response_factor,
    Risk_factor
  FROM
    ReviewDataSet
);
SELECT * FROM ReviewDataSet_Format;
```

DA228- BigData Systems Assignment-5

```
1 %sql
2
3 CREATE OR REPLACE TEMPORARY VIEW ReviewDataSet_Format AS
4 (
5     SELECT
6         Rest_Id,
7         Rest_Name,
8         Address,
9         City,
10        State,
11        Zipcode,
12        Latitude,
13        Longitude,
14        Inspection_ID,
15        TO_DATE(RIGHT(Inspection_date, 8), "yyyyMMdd") AS Inspection_date,
16        TO_TIMESTAMP(Inspection_Time) AS Inspection_Time,
17        Points,
18        Inspection_Category,
19        RIGHT(Response_ID,CHARINDEX('_', (REVERSE(Response_ID))) - 1) AS Response_ID,
20        Response_factor,
21        Risk_factor
22    FROM
23        ReviewDataSet
24 );
25 SELECT * FROM ReviewDataSet_Format;
26
```

▶ (1) Spark Jobs

	Rest_Id	Rest_Name	Address	City	State	Zipcode	Latitude	Longitude
1	2764	AL-HAMRA	3083 16th St	San Francisco	CA	94103	37.76491	-122.42135
2	1154	SUNFLOWER RESTAURANT	506 Valencia St	San Francisco	CA	94103	37.76468	-122.421906
3	69665	Shalimar Restaurant	532 Jones St	San Francisco	CA	94102	null	null
4	1154	SUNFLOWER RESTAURANT	506 Valencia St	San Francisco	CA	94103	37.76468	-122.421906
5	1154	SUNFLOWER RESTAURANT	506 Valencia St	San Francisco	CA	94103	37.76468	-122.421906
6	2749	TAQUERIA CANCUN	3211 MISSION St	San Francisco	CA	94110	37.745434	-122.419945
7	2638	SF BAGEL CO. (KATZ BAGELS)	3147 16th St	San Francisco	CA	94103	37.764812	-122.42309

Truncated results, showing first 1000 rows.



Displaying Descriptive Stats

```
df_RDformat =spark.sql("SELECT * FROM ReviewDataSet_Format")
display(df_RDformat.describe())
```

Cmd 5

```
1 df_RDformat =spark.sql("SELECT * FROM ReviewDataSet_Format")
2 display(df_RDformat.describe())
```





DA228- BigData Systems Assignment-5

```
1 df_RDformat =spark.sql("SELECT * FROM ReviewDataSet_Format")
2 display(df_RDformat.describe())
```

► (2) Spark Jobs

	summary	Rest_Id	Rest_Name	Address	City	State	Zipcode	Latitude	Longi
1	count	51731	51731	51731	51731	51731	50494	29189	29189
2	mean	50675.57279000986	1588.6	377.0	null	null	317792.5848043586	37.7434784244497	-122.3
3	stddev	35443.47046923853	383.26205134346395	0.0	null	null	1.4507252683961099E7	1.0368811054803597	3.3600
4	min	19	100% Dessert Cafe	001 WEST PORTAL Ave	San Francisco	CA	0	0.0	-122.5
5	max	98377	vive la tarte	Various Farmers Markets	San Francisco	CA	Ca	37.824493	0.0

Showing all 5 rows.

Displaying summary

```
display(df_RDformat.summary())
```





Cmd 6

```
1 display(df_RDformat.summary())
```

► (2) Spark Jobs

	summary	Rest_Id	Rest_Name	Address	City	State	Zipcode	Latitude	Longi
1	count	51731	51731	51731	51731	51731	50494	29189	29189
2	mean	50675.57279000986	1588.6	377.0	null	null	317792.5848043586	37.7434784244497	-122.3
3	stddev	35443.47046923853	383.26205134346395	0.0	null	null	1.4507252683961099E7	1.0368811054803597	3.3600
4	min	19	100% Dessert Cafe	001 WEST PORTAL Ave	San Francisco	CA	0	0.0	-122.5
5	25%	5855	1760.0	377.0	null	null	94107.0	37.755283	-122.4
6	50%	66191	1760.0	377.0	null	null	94111.0	37.78035	-122.4
7	75%	82218	1760.0	377.0	null	null	94121.0	37.789516	-122.4

Showing all 8 rows.

Displaying the distinct counts

```
print('Count of rows: {0}'.format(df_RDformat.count()))
print('Count of distinct rows: {0}'.format(df_RDformat.distinct().count()))
```

```
1 print('Count of rows: {0}'.format(df_RDformat.count()))
2 print('Count of distinct rows: {0}'.format(df_RDformat.distinct().count()))
```

► (5) Spark Jobs

Count of rows: 51731

Count of distinct rows: 51684

DA228- BigData Systems Assignment-5

Removing the duplicates

```
df_RDformat = df_RDformat.dropDuplicates()
display(df_RDformat.select("Latitude").describe())
display(df_RDformat.select("Longitude").describe())
display(df_RDformat.select("Inspection_Points").describe())
```

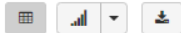
Cmd 7

```
1 # Removing duplicate values and showing descriptive statistics for numerical columns.
2 df_RDformat = df_RDformat.dropDuplicates()
3 display(df_RDformat.select("Latitude").describe())
4 display(df_RDformat.select("Longitude").describe())
5 display(df_RDformat.select("Points").describe())
6 |
```

► (9) Spark Jobs

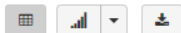
	summary ▲	Latitude ▲
1	count	29168
2	mean	37.74346717909281
3	stddev	1.0372540154265812
4	min	0.0
5	max	37.824493

Showing all 5 rows.

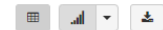


	summary ▲	Longitude ▲
1	count	29168
2	mean	-122.33574217881596
3	stddev	3.361223087152367
4	min	-122.510895
5	max	0.0

Showing all 5 rows.

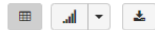


Showing all 5 rows.



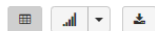
	summary ▲	Longitude ▲
1	count	29168
2	mean	-122.33574217881596
3	stddev	3.361223087152367
4	min	-122.510895
5	max	0.0

Showing all 5 rows.



	summary ▲	Inspection_Points ▲
1	count	38166
2	mean	85.94783314992402
3	stddev	8.780260150847441
4	min	45
5	max	100

Showing all 5 rows.



DA228- BigData Systems Assignment-5

Finding and displaying the Missing values

```
import pyspark.sql.functions as fn
df_RDformat.agg(*[
    (1 - (fn.count(c) / fn.count('*'))).alias(c + '_missing')
    for c in df_RDformat.columns
]).display()
```

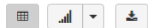
Cmd 8

```
1 # Finding and displaying the Missing values
2 import pyspark.sql.functions as fn
3 df_RDformat.agg(*[
4     (1 - (fn.count(c) / fn.count('*'))).alias(c + '_missing')
5     for c in df_RDformat.columns
6 ]).display()
7
```

▶ (3) Spark Jobs

	Rest_Id_missing ▲	Rest_Name_missing ▲	Address_missing ▲	City_missing ▲	State_missing ▲	Zipcode_missing ▲	Latitude_missing ▲	Longitude_missing ▲	Ins ▲
1	0	0	0	0	0	0.023914557696772643	0.4356473957124062	0.4356473957124062	0.6

Showing all 1 rows.



Replacing missing values with mean

%sql

-- Finding mean for Latitude

select mean(Latitude) from ReviewDataSet_Format

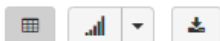
Cmd 14

```
1 %sql
2 -- Finding mean for Latitude
3 select mean(Latitude) from ReviewDataSet_Format
```

▶ (2) Spark Jobs

	mean(Latitude) ▲
1	37.7434784244497

Showing all 1 rows.



DA228- BigData Systems Assignment-5

%sql

--Filling missing Latitude values with it's mean value

select Rest_Id ,coalesce(Latitude ,(select avg(Latitude) from ReviewDataSet_Format)) from ReviewDataSet_Format

```
1 %sql
2 --Filling missing Latitude values with it's mean value
3 select Rest_Id ,coalesce(Latitude ,(select avg(Latitude) from ReviewDataSet_Format) ) from ReviewDataSet_Format
```

▶ (3) Spark Jobs

	Rest_Id	coalesce(CAST(Latitude AS DOUBLE), scalarsubquery())
1	2764	37.76491165161133
2	1154	37.764678955078125
3	69665	37.7434784244497
4	1154	37.764678955078125
5	1154	37.764678955078125
6	2749	37.74543380737305
7	2638	37.76481246948242

Truncated results, showing first 1000 rows.



Checking for skewness

df_RDformat.agg({'Latitude': 'skewness'}).show()

df_RDformat.agg({'Longitude': 'skewness'}).show()

df_RDformat.agg({'Inspection_Points': 'skewness'}).show()

Cmd 10

```
1 #Checking for skewness
2 df_RDformat.agg({'Latitude': 'skewness'}).show()
3 df_RDformat.agg({'Longitude': 'skewness'}).show()
4 df_RDformat.agg({'Points': 'skewness'}).show()
```

▶ (9) Spark Jobs

```
+-----+
|skewness(Latitude)|
+-----+
|-36.34190678595643|
+-----+
```

```
+-----+
|skewness(Longitude)|
+-----+
| 36.36688577115409|
+-----+
```

```
+-----+
|skewness(Inspection_Points)|
+-----+
| -0.791032609569868|
+-----+
```


Finding the Correlations between numerical values

#Finding the Correlation

```
numcol = ['Latitude', 'Longitude', 'Inspection_Points']  
desc = df_RDformat.describe(numcol)  
desc.show()
```

```
n_numerical = len(numcol)
```

```
correlation = []
```

```
for i in range(0, n_numerical):
```

```
    temp = [None] * i
```

```
    for j in range(i, n_numerical):
```

```
        temp.append(df_RDformat.correlation(numcol[i], numcol[j]))
```

```
    correlation.append(temp)
```

correlation

Cmd 11

```
1  #Finding the Correlations between numerical values  
2  
3  df_RDformat.corr('Latitude', 'Longitude')  
4  #df_format.corr('Longitude', 'Latitude')
```

Cmd 12

```
1  #Correlations matrix  
2  n_numerical = len(numcol)  
3  
4  corr = []  
5  
6  for i in range(0, n_numerical):  
7      temp = [None] * i  
8  
9      for j in range(i, n_numerical):  
10         temp.append(df_RDformat.corr(numcol[i], numcol[j]))  
11         corr.append(temp)  
12  
13  corr
```

DA228- BigData Systems Assignment-5

► (15) Spark Jobs

summary	Latitude	Longitude	Inspection_Points
count	29168	29168	38166
mean	37.74346717909281	-122.33574217881596	85.94783314992402
stddev	1.0372540154265812	3.361223087152367	8.780260150847441
min	0.0	-122.510895	45
max	37.824493	0.0	100

```
[[1.0, -0.999999393746342, 0.13241015021497107],  
 [None, 1.0, -0.13240964485822712],  
 [None, None, 1.0]]
```

Finding number of distinct restaurants in each zipcode.

%sql

--Find number of distinct restaurants in each zipcode.

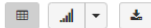
select count(distinct Rest_Id), Zipcode from ReviewDataSet_Format group by Zipcode ;

```
Cmd 12  
1 %sql  
2 --Find number of distinct restaurants in each zipcode.  
3 select count( distinct Rest_Id), Zipcode from ReviewDataSet_Format group by Zipcode ;
```

► (3) Spark Jobs

	count(DISTINCT Rest_Id)	Zipcode
1	440	94102
2	392	94107
3	1	Ca
4	1	94621
5	145	94104
6	1	95122
7	49	94131

Showing all 54 rows.



DA228- BigData Systems Assignment-5

%sql

select distinct Zipcode , Rest_Name from ReviewDataSet_Format order by Zipcode , Rest_Name ;

Cmd 13

```
1 %sql
2 select distinct Zipcode , Rest_Name from ReviewDataSet_Format order by Zipcode , Rest_Name ;
```

► (2) Spark Jobs

	Zipcode ▲	Rest_Name ▲
1	null	1760
2	null	24th and Folsom Eatery
3	null	3Geeks
4	null	94637 Doggie Diner/Great House Beer
5	null	94654 Derby Grill/Tony's Pizza
6	null	94884 Great House of Brews
7	null	94971 Gilroy Garlic Stand

Truncated results, showing first 1000 rows.

