

Big Data Analysis

Data analysis with Pyspark

Kunal Khurana

2024-03-16

Table of contents

Installation	3
Installing libraries	3
Preprocessing	4
Visualization	9

Installation

```
!pip install pyspark
```

Collecting pyspark

Downloading pyspark-3.5.1.tar.gz (317.0 MB)

317.0/317.0 MB 2.8 MB/s eta 0:00:00

Preparing metadata (setup.py) ... done

Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.10/dist-packages (from pyspark)

Building wheels for collected packages: pyspark

Building wheel for pyspark (setup.py) ... done

Created wheel for pyspark: filename=pyspark-3.5.1-py2.py3-none-any.whl size=317488491 sha256=...

Stored in directory: /root/.cache/pip/wheels/80/1d/60/2c256ed38dddce2fdd93be545214a63e02fb...

Successfully built pyspark

Installing collected packages: pyspark

Successfully installed pyspark-3.5.1

Installing libraries

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import count, desc, col, max, struct
import matplotlib.pyplot as plt
```

```
spark = SparkSession.builder.appName('spark_app').getOrCreate()
```

```
traffic_collision_data = '/content/traffic-collision-data-from-2010-to-present.csv'
```

```
# prompt: load traffic_collision_data with spark
```

```
df = spark.read.csv(traffic_collision_data, header=True, inferSchema=True)
```

```
df.show()
```

DR Number	Date Reported	Date Occurred	Time Occurred	Area ID	Area Name	Report
191323054	2019-11-30 00:00:00	2019-11-30T00:00:...		0130	13	Newton
192020666	2019-11-30 00:00:00	2019-11-30T00:00:...		0015	20	Olympic
191616992	2019-11-30 00:00:00	2019-11-30T00:00:...		0230	16	Foothill
191824082	2019-11-30 00:00:00	2019-11-30T00:00:...		0730	18	Southeast
191616980	2019-11-30 00:00:00	2019-11-30T00:00:...		0720	16	Foothill
191824078	2019-11-30 00:00:00	2019-11-30T00:00:...		1050	18	Southeast
190417458	2019-11-30 00:00:00	2019-11-30T00:00:...		0130	04	Hollenbeck
191616985	2019-11-30 00:00:00	2019-11-30T00:00:...		0700	16	Foothill
191718751	2019-11-30 00:00:00	2019-11-30T00:00:...		1230	17	Devonshire
191718743	2019-11-30 00:00:00	2019-11-30T00:00:...		0010	17	Devonshire
191824080	2019-11-30 00:00:00	2019-11-30T00:00:...		0945	18	Southeast
190720157	2019-11-29 00:00:00	2019-11-29T00:00:...		1115	07	Wilshire
190518783	2019-11-29 00:00:00	2019-11-29T00:00:...		0650	05	Harbor
192119165	2019-11-29 00:00:00	2019-11-29T00:00:...		1005	21	Topanga
191018309	2019-11-29 00:00:00	2019-11-29T00:00:...		0710	10	West Valley
192119188	2019-11-29 00:00:00	2019-11-29T00:00:...		1800	21	Topanga
191018335	2019-11-29 00:00:00	2019-11-29T00:00:...		1600	10	West Valley
191616964	2019-11-29 00:00:00	2019-11-29T00:00:...		0645	16	Foothill
190222525	2019-11-29 00:00:00	2019-11-29T00:00:...		0545	02	Rampart
190518807	2019-11-30 00:00:00	2019-11-29T00:00:...		2220	05	Harbor

only showing top 20 rows

Preprocessing

```
# prompt: drop the DR number column

df = df.drop('DR Number')

df.show()
```

Date Reported	Date Occurred	Time Occurred	Area ID	Area Name	Reporting District
2019-11-30 00:00:00	2019-11-30T00:00:...		0130	13	Newton
2019-11-30 00:00:00	2019-11-30T00:00:...		0015	20	Olympic
2019-11-30 00:00:00	2019-11-30T00:00:...		0230	16	Foothill


```
|-- Victim Descent: string (nullable = true)
|-- Premise Code: string (nullable = true)
|-- Premise Description: string (nullable = true)
|-- Address: string (nullable = true)
|-- Cross Street: string (nullable = true)
|-- Location: string (nullable = true)
|-- Zip Codes: string (nullable = true)
|-- Census Tracts: string (nullable = true)
|-- Precinct Boundaries: string (nullable = true)
|-- LA Specific Plans: string (nullable = true)
|-- Council Districts: string (nullable = true)
|-- Neighborhood Councils (Certified): string (nullable = true)
```

```
# prompt: see the shape of a dataframe
```

```
print(f"Number of rows: {df.count()}")
print(f"Number of columns: {len(df.columns)}")
```

```
Number of rows: 294684
```

```
Number of columns: 23
```

```
# prompt: select two columns area id and victim age
```

```
df.select('Area ID', 'Victim Age').show()
```

```
+-----+-----+
|Area ID|Victim Age|
+-----+-----+
|    13|      NULL|
|    20|        40|
|    16|        18|
|    18|        23|
|    16|      NULL|
|    18|        54|
|    04|        33|
|    16|        35|
|    17|        51|
|    17|        23|
|    18|        26|
|    07|        17|
```

```

|    05|    44|
|    21|    58|
|    10|    99|
|    21|    22|
|    10|    29|
|    16|    28|
|    02|    38|
|    05|    44|
+-----+-----+

```

only showing top 20 rows

```
# prompt: select those records where the victim age is 17
```

```
df.select('Area ID', 'Victim Age').where(df['Victim Age'] == 17).show()
```

```

+-----+-----+
|Area ID|Victim Age|
+-----+-----+
|    07|    17|
|    09|    17|
|    09|    17|
|    03|    17|
|    19|    17|
|    20|    17|
|    15|    17|
|    03|    17|
|    17|    17|
|    21|    17|
|    03|    17|
|    17|    17|
|    08|    17|
|    13|    17|
|    21|    17|
|    13|    17|
|    09|    17|
|    07|    17|
|    15|    17|
|    06|    17|
+-----+-----+

```

only showing top 20 rows

```
df.select('*').where(df['Victim Age'] == 17).show()
```

Date Reported	Date Occurred	Time Occurred	Area ID	Area Name	Reporting District
2019-11-29 00:00:00	2019-11-29T00:00:...	1115	07	Wilshire	07
2019-11-29 00:00:00	2019-11-28T00:00:...	2341	09	Van Nuys	09
2019-11-20 00:00:00	2019-11-20T00:00:...	1640	09	Van Nuys	09
2019-11-18 00:00:00	2019-11-18T00:00:...	1625	03	Southwest	03
2019-11-15 00:00:00	2019-11-15T00:00:...	1820	19	Mission	19
2019-11-14 00:00:00	2019-11-14T00:00:...	2222	20	Olympic	20
2019-11-14 00:00:00	2019-11-14T00:00:...	0935	15	N Hollywood	15
2019-11-11 00:00:00	2019-11-11T00:00:...	1500	03	Southwest	03
2019-05-09 00:00:00	2019-05-09T00:00:...	2030	17	Devonshire	17
2019-05-09 00:00:00	2019-05-09T00:00:...	2000	21	Topanga	21
2019-05-05 00:00:00	2019-05-05T00:00:...	1240	03	Southwest	03
2019-04-28 00:00:00	2019-04-28T00:00:...	2040	17	Devonshire	17
2019-04-26 00:00:00	2019-04-26T00:00:...	2030	08	West LA	08
2019-11-06 00:00:00	2019-11-06T00:00:...	0700	13	Newton	13
2019-11-07 00:00:00	2019-11-06T00:00:...	1740	21	Topanga	21
2019-11-04 00:00:00	2019-11-04T00:00:...	1220	13	Newton	13
2019-11-01 00:00:00	2019-11-01T00:00:...	1645	09	Van Nuys	09
2019-10-29 00:00:00	2019-10-29T00:00:...	0810	07	Wilshire	07
2019-10-27 00:00:00	2019-10-27T00:00:...	0945	15	N Hollywood	15
2019-10-27 00:00:00	2019-10-27T00:00:...	1305	06	Hollywood	06

only showing top 20 rows

```
# prompt: find out which area is most prone to crimes
```

```
most_crime_prone_area = df.groupBy('Area ID').agg(count('*').alias('total_crimes')).orderBy('total_crimes', descending=True).first()
print(f"Most crime prone area: {most_crime_prone_area}")
```

Most crime prone area: 12

```
# prompt: find out top 10 crime codes
```

```
top_10_crime_codes = df.groupBy('Crime Code Description').agg(count('*').alias('total_crimes')).orderBy('total_crimes', descending=True).limit(10)
```



```
print("Top 10 crime codes:")
for code in top_10_crime_codes:
    print(f"\t- {code}")
```

Top 10 crime codes:

- TRAFFIC COLLISION
- 180

```
# prompt: find out top 10 victim age

top_10_victim_ages = df.groupby('Victim Age').agg(count('*').alias('total_victims')).order

print("Top 10 victim ages:")
for age in top_10_victim_ages:
    print(f"\t- {age}")
```

Top 10 victim ages:

- None
- 30
- 25
- 27
- 24
- 28
- 26
- 23
- 35
- 29

Visualization

```
# prompt: visualize the result of top 10 victim age in the form of a pie chart

# Get the top 10 victim ages and their counts
top_10_victim_ages = df.groupby('Victim Age').agg(count('*').alias('total_victims')).order

# Create a pie chart of the top 10 victim ages
plt.figure(figsize=(12, 6))
plt.pie(top_10_victim_ages['total_victims'], labels=top_10_victim_ages['Victim Age'], auto
```

```
plt.title('Top 10 Victim Ages')  
plt.show()
```

