# Chicago_taxi

February 16, 2024

## 1 - BigData

### 1.1 :

10 , . PySpark , Docker .

, 2022 2023 . , 15 , . , , , , .

**Trip ID:** .

**Taxi ID:** .

**Trip Start Timestamp:** .

**Trip End Timestamp:** .

**Trip Seconds:** .

**Trip Miles:** .

**Pickup Census Tract:** .

**Dropoff Census Tract:** .

**Pickup Community Area:** , .

**Dropoff Community Area:** , .

**Fare:** .

**Tips:** .

**Tolls:** .

**Extras:** .

**Trip Total:** .

**Payment Type:** .

**Company:** , .

**Pickup Centroid Latitude:** .

**Pickup Centroid Longitude:** .

**Pickup Centroid Location:** .

**Dropoff Centroid Latitude:** .

**Dropoff Centroid Longitude:** .

**Dropoff Centroid Location:** .

### 1.2

**1.** : - .

**2.** : - , . , .

**3.** : - , . .

**4.** : - (EDA) .

**5.** : - , , .

**6.** : - . .

**7.** : - .

**8.** : - , . .

[1]:

```
#
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import warnings
import seaborn as sns
import folium
from folium.plugins import HeatMap

#      PySpark
from pyspark.sql import SparkSession
from pyspark.sql import functions as F
from pyspark.sql.types import DoubleType, IntegerType, TimestampType
from pyspark.sql.window import Window
from pyspark.sql.functions import to_timestamp, col, isnan, count, round,␣
 ↪countDistinct
from pyspark.ml.feature import StringIndexer, OneHotEncoder, VectorAssembler,␣
 ↪StandardScaler
from pyspark.ml.regression import RandomForestRegressor, DecisionTreeRegressor,␣
 ↪LinearRegression
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml import Pipeline
from pyspark.sql.functions import col, to_date, hour

#
warnings.filterwarnings('ignore')
```

[2]:
```
#      SparkSession       "TaxiDemandPrediction"
spark = SparkSession.builder.appName("TaxiDemandPrediction").getOrCreate()
```

## 1.3

CSV- , “Taxi_Trips_-2022.csv” “Taxi_Trips-_2023.csv”, Apache Spark. , “LEGACY”.

[3]:
```
taxi_spark_2022 = spark.read.csv('Taxi_Trips_-_2022.csv', header=True,␣
 ↪inferSchema = True)
taxi_spark_2023 = spark.read.csv('Taxi_Trips_-_2023.csv', header=True,␣
 ↪inferSchema = True)
```

[4]:
```
spark.conf.set("spark.sql.legacy.timeParserPolicy", "LEGACY")
```

[5]:
```
taxi_spark_2022
```

[5]:
```
DataFrame[Trip ID: string, Taxi ID: string, Trip Start Timestamp: string, Trip
End Timestamp: string, Trip Seconds: int, Trip Miles: double, Pickup Census
Tract: bigint, Dropoff Census Tract: bigint, Pickup Community Area: int, Dropoff
Community Area: int, Fare: double, Tips: double, Tolls: double, Extras: double,
Trip Total: double, Payment Type: string, Company: string, Pickup Centroid
Latitude: double, Pickup Centroid Longitude: double, Pickup Centroid Location:
string, Dropoff Centroid Latitude: double, Dropoff Centroid Longitude: double,
Dropoff Centroid  Location: string]
```

[6]:
```
taxi_spark_2023
```

[6]:
```
DataFrame[Trip ID: string, Taxi ID: string, Trip Start Timestamp: string, Trip
End Timestamp: string, Trip Seconds: int, Trip Miles: double, Pickup Census
Tract: bigint, Dropoff Census Tract: bigint, Pickup Community Area: int, Dropoff
Community Area: int, Fare: double, Tips: double, Tolls: double, Extras: double,
Trip Total: double, Payment Type: string, Company: string, Pickup Centroid
Latitude: double, Pickup Centroid Longitude: double, Pickup Centroid Location:
string, Dropoff Centroid Latitude: double, Dropoff Centroid Longitude: double,
Dropoff Centroid  Location: string]
```

[7]:
```
#
pd.set_option('display.max_columns', None)

#       describe()    DataFrame
taxi_2022 = taxi_spark_2022.describe().toPandas()
```

```python
#
print("                taxi_spark_2022:")
print(taxi_2022)
```

```
                taxi_spark_2022:
  summary                                   Trip ID  \
0   count                                   6382425
1    mean                                      None
2  stddev                                      None
3     min  000000bb18f0563c13ad977fc05b901474cd3941
4     max  ffffff1aae5322736637e16dd2faecb5dfebe81a

                                             Taxi ID    Trip Start Timestamp  \
0                                            6382425                 6382425
1                                               None                    None
2                                               None                    None
3  0041f8f0c91881c1e1913f2548522495fe3c4c719aa67f…  01/01/2022 01:00:00 AM
4  fff84aa08ac78890c6e7da64b817cbd9aad6a124104e09…  12/31/2022 12:45:00 PM

       Trip End Timestamp        Trip Seconds         Trip Miles  \
0                 6382213             6380960            6382369
1                    None  1198.2085212883328  6.185568905527588
2                    None   1895.664878082732     8.002858369488
3  01/01/2022 01:00:00 AM                   0                0.0
4  12/31/2022 12:45:00 PM               86341            2967.54

     Pickup Census Tract  Dropoff Census Tract Pickup Community Area  \
0                2623831               2675331               5868572
1  1.7031468160376106E10  1.703141184686421E10     32.35048253646713
2      368945.9010693637    345773.49235842755    25.203045304909356
3            17031010100           17031010100                     1
4            17031980100           17031980100                    77

  Dropoff Community Area                Fare                Tips  \
0                5748741             6378889             6378889
1       25.8431748795084   21.72931312020104  2.7545550142038486
2     20.925425235069905  49.416238999460845    4.08389167014634
3                      1                 0.0                 0.0
4                     77             9999.75               496.0

                 Tolls             Extras          Trip Total Payment Type  \
0              6378889            6378889             6378889      6382425
1  0.02128382074057096  2.163035586604502   26.82509761809589         None
2    7.659938846744798  21.75269211485504  56.964604407228194         None
3                  0.0                0.0                 0.0         Cash
4              6666.66            8888.88             9999.75      Unknown
```

```
         Company Pickup Centroid Latitude Pickup Centroid Longitude  \
0        6382425                  5870874                   5870874
1           None       41.899921551854426        -87.68816038912836
2           None      0.06015143048421792       0.10469957717265778
3  24 Seven Taxi             41.651921576             -87.913624596
4      U Taxicab             42.021223593             -87.530712484

               Pickup Centroid Location Dropoff Centroid Latitude  \
0                               5870874                   5784494
1                                  None         41.89471203364863
2                                  None       0.05620670234136166
3  POINT (-87.5307124836 41.7030053028)              41.660136051
4   POINT (-87.913624596 41.9802643146)              42.021223593

  Dropoff Centroid Longitude            Dropoff Centroid  Location
0                    5784494                               5784494
1         -87.66248676270533                                  None
2         0.0733198539959098                                  None
3              -87.913624596  POINT (-87.5313862567 41.7204632831)
4              -87.531386257   POINT (-87.913624596 41.9802643146)
```

[8]:
```
#      describe()   DataFrame
taxi_2023 = taxi_spark_2023.describe().toPandas()

#
print("                taxi_spark_2023:")
taxi_2023
```

```
                taxi_spark_2023:
```

[8]:
```
  summary                                   Trip ID  \
0    count                                  3783730
1     mean                                     None
2   stddev                                     None
3      min  0000012deb83dbb55726d5a75c374197d0641fa0
4      max  fffffe03acfa1552c98fad12d73ff0aca70a5c2a

                                             Taxi ID    Trip Start Timestamp  \
0                                            3783730                 3783730
1                                               None                    None
2                                               None                    None
3  00110971c7c4a7173fcf93f49a22d6b9b0a02c27c4b9f8…  01/01/2023 01:00:00 AM
4  ffd231d2536b9463d888cfbb42f36d543b37d22d96a6dd…  08/01/2023 12:00:00 AM

       Trip End Timestamp        Trip Seconds         Trip Miles  \
0                 3783682             3783012            3783717
```

```
1                   None  1235.3278387697421  6.471233194766421
2                   None  1736.5661018295184  7.593310441212216
3  01/01/2023 01:00:00 AM                   0                0.0
4  10/17/2022 10:00:00 AM               86340              945.4

     Pickup Census Tract   Dropoff Census Tract Pickup Community Area  \
0                1650232                1617623               3615963
1  1.7031501647110167E10  1.7031414613682808E10     35.03091652209937
2     373503.06470611464     344107.34525212087      26.0787201470195
3            17031010100            17031010100                     1
4            17031980100            17031980100                    77

  Dropoff Community Area                Fare               Tips  \
0                3419046             3778327            3778327
1     26.097845714857304  21.995519501091266  2.926380662658419
2      20.91789151274584  22.233947201108272  4.200411938894532
3                      1                 0.0                0.0
4                     77             9999.75              375.0

                 Tolls             Extras          Trip Total Payment Type  \
0              3778327            3778327             3778327      3783730
1  0.05000318394887473  2.241853884007393  27.371134848836103         None
2   11.569869455292247  19.47903838213312   37.03373571024758         None
3                  0.0                0.0                 0.0         Cash
4              6666.66            9446.65             9999.75      Unknown

                   Company Pickup Centroid Latitude Pickup Centroid Longitude  \
0                  3783730                  3617351                   3617351
1                     None        41.90200067412448        -87.69906175284667
2                     None      0.06251123711511891       0.11213128488994714
3  2733 - 74600 Benny Jona             41.651921576             -87.913624596
4                U Taxicab             42.021223593             -87.531386257

               Pickup Centroid Location Dropoff Centroid Latitude  \
0                               3617351                   3441935
1                                  None         41.89419484402875
2                                  None       0.05656711663584079
3  POINT (-87.5313862567 41.7204632831)              41.660136051
4   POINT (-87.913624596 41.9802643146)              42.021223593

  Dropoff Centroid Longitude           Dropoff Centroid  Location
0                    3441935                              3441935
1         -87.66236339478645                                 None
2        0.07345541720850336                                 None
3              -87.913624596  POINT (-87.5349029012 41.707311449)
4              -87.534902901  POINT (-87.913624596 41.9802643146)
```

[9]:

```
#
taxi_spark = taxi_spark_2022.union(taxi_spark_2023)
```

[10]:

```
taxi_spark
```

[10]:

```
DataFrame[Trip ID: string, Taxi ID: string, Trip Start Timestamp: string, Trip
End Timestamp: string, Trip Seconds: int, Trip Miles: double, Pickup Census
Tract: bigint, Dropoff Census Tract: bigint, Pickup Community Area: int, Dropoff
Community Area: int, Fare: double, Tips: double, Tolls: double, Extras: double,
Trip Total: double, Payment Type: string, Company: string, Pickup Centroid
Latitude: double, Pickup Centroid Longitude: double, Pickup Centroid Location:
string, Dropoff Centroid Latitude: double, Dropoff Centroid Longitude: double,
Dropoff Centroid  Location: string]
```

[11]:

```
type(taxi_spark)
```

[11]:

```
pyspark.sql.dataframe.DataFrame
```

[12]:

```
taxi_spark.columns
```

[12]:

```
['Trip ID',
 'Taxi ID',
 'Trip Start Timestamp',
 'Trip End Timestamp',
 'Trip Seconds',
 'Trip Miles',
 'Pickup Census Tract',
 'Dropoff Census Tract',
 'Pickup Community Area',
 'Dropoff Community Area',
 'Fare',
 'Tips',
 'Tolls',
 'Extras',
 'Trip Total',
 'Payment Type',
 'Company',
 'Pickup Centroid Latitude',
 'Pickup Centroid Longitude',
 'Pickup Centroid Location',
 'Dropoff Centroid Latitude',
 'Dropoff Centroid Longitude',
 'Dropoff Centroid  Location']
```

[13]:

```
taxi_spark.count()
```

[13]:

```
10166155
```

[14]:
```
#
missing_data = taxi_spark.select([F.count(F.when(F.isnan(c) | F.col(c).
 ↪isNull(), c)).alias(c) for c in taxi_spark.columns])
missing_data.toPandas()
```

[14]:
```
   Trip ID  Taxi ID  Trip Start Timestamp  Trip End Timestamp  Trip Seconds  \
0        0        0                     0                 260          2183

   Trip Miles  Pickup Census Tract  Dropoff Census Tract  \
0          69              5892092               5873201

   Pickup Community Area  Dropoff Community Area  Fare  Tips  Tolls  Extras  \
0                 681620                  998368  8939  8939   8939    8939

   Trip Total  Payment Type  Company  Pickup Centroid Latitude  \
0        8939             0        0                    677930

   Pickup Centroid Longitude  Pickup Centroid Location  \
0                     677930                    677930

   Dropoff Centroid Latitude  Dropoff Centroid Longitude  \
0                     939726                      939726

   Dropoff Centroid  Location
0                      939726
```

[15]:
```
#
missing_data_p = missing_data.select(*((F.round(F.col(row) / taxi_spark.count()␣
 ↪* 100, 1)).alias(row) for row in missing_data.columns))
missing_data_p.toPandas()
```

[15]:
```
   Trip ID  Taxi ID  Trip Start Timestamp  Trip End Timestamp  Trip Seconds  \
0      0.0      0.0                   0.0                 0.0           0.0

   Trip Miles  Pickup Census Tract  Dropoff Census Tract  \
0         0.0                 58.0                  57.8

   Pickup Community Area  Dropoff Community Area  Fare  Tips  Tolls  Extras  \
0                    6.7                     9.8   0.1   0.1    0.1     0.1

   Trip Total  Payment Type  Company  Pickup Centroid Latitude  \
0         0.1           0.0      0.0                       6.7

   Pickup Centroid Longitude  Pickup Centroid Location  \
0                        6.7                       6.7

   Dropoff Centroid Latitude  Dropoff Centroid Longitude  \
0                         9.2                         9.2

   Dropoff Centroid  Location
0                          9.2
```

, 0%, ,
.

### 1.4

[16]:
```
taxi_spark.printSchema()
```

```
root
 |-- Trip ID: string (nullable = true)
 |-- Taxi ID: string (nullable = true)
 |-- Trip Start Timestamp: string (nullable = true)
 |-- Trip End Timestamp: string (nullable = true)
 |-- Trip Seconds: integer (nullable = true)
 |-- Trip Miles: double (nullable = true)
 |-- Pickup Census Tract: long (nullable = true)
 |-- Dropoff Census Tract: long (nullable = true)
 |-- Pickup Community Area: integer (nullable = true)
 |-- Dropoff Community Area: integer (nullable = true)
 |-- Fare: double (nullable = true)
 |-- Tips: double (nullable = true)
 |-- Tolls: double (nullable = true)
 |-- Extras: double (nullable = true)
 |-- Trip Total: double (nullable = true)
 |-- Payment Type: string (nullable = true)
 |-- Company: string (nullable = true)
 |-- Pickup Centroid Latitude: double (nullable = true)
 |-- Pickup Centroid Longitude: double (nullable = true)
 |-- Pickup Centroid Location: string (nullable = true)
 |-- Dropoff Centroid Latitude: double (nullable = true)
 |-- Dropoff Centroid Longitude: double (nullable = true)
 |-- Dropoff Centroid  Location: string (nullable = true)
```

printSchema() DataFrame . ,
“string” ( ), , , “Trip Seconds”, “Trip Miles”, “Fare”, “Tips”
, .

[17]:
```
taxi_spark.show()
```

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

```
-------------------+-------------------------+--------------------------+-------
-------------------+
|             Trip ID|             Taxi ID|Trip Start Timestamp|  Trip End
Timestamp|Trip Seconds|Trip Miles|Pickup Census Tract|Dropoff Census
Tract|Pickup Community Area|Dropoff Community Area| Fare|Tips|Tolls|Extras|Trip
Total|Payment Type|             Company|Pickup Centroid Latitude|Pickup Centroid
Longitude|Pickup Centroid Location|Dropoff Centroid Latitude|Dropoff Centroid
Longitude|Dropoff Centroid  Location|
+--------------------+--------------------+--------------------+----------------
----+------------+----------+-------------------+--------------------+----------
-----------+----------------------+-----+----+-----+------+----------+----------
--+--------------------+------------------------+-------------------------+-----
-------------------+-------------------------+--------------------------+-------
-------------------+
|bcfa19f2539021c05…|368ce5511598af2cc…|01/01/2022 12:00:…|01/01/2022
12:00:…|         152|       0.1|               null|                null|
null|                  null| 3.75| 0.0|  0.0|   0.0|      3.75|        Cash|
Medallion Leasin|                    null|                     null|
null|                     null|                      null|
null|
|2aba69ff015f9ea8e…|449fa490955275713…|01/01/2022 12:00:…|01/01/2022
12:30:…|        2360|     17.44|               null|                null|
null|                     8|47.75| 0.0|  0.0|   5.0|     52.75|        Cash|
Flash Cab|                    null|                     null|
null|             41.899602111|             -87.633308037|      POINT
(-87.633308…|
|54d812a0b88f8f970…|f98ae5e71fdda8806…|01/01/2022 12:00:…|01/01/2022
12:00:…|         536|      4.83|               null|                null|
28|                    22|14.75| 0.0|  0.0|   0.0|     14.75|        Cash|
Globe Taxi|            41.874005383|             -87.66351755|    POINT
(-87.663517…|              41.92276062|             -87.699155343|      POINT
(-87.699155…|
|7125b9e03a0f16c2d…|8eca35a570101ad24…|01/01/2022 12:00:…|01/01/2022
12:15:…|         897|      2.07|               null|                null|
8|                    32| 9.75| 0.0|  0.0|   1.5|     11.25|        Cash|
Sun Taxi|            41.899602111|            -87.633308037|    POINT
(-87.633308…|             41.878865584|             -87.625192142|      POINT
(-87.625192…|
|f1a650ee419b4e52d…|e2d8418fcdb061eee…|01/01/2022 12:00:…|01/01/2022
12:30:…|        2200|      2.48|               null|                null|
8|                    32| 9.36|2.14|  0.0|   0.0|      11.5|      Mobile|Chicago
Independents|            41.899602111|            -87.633308037|    POINT
(-87.633308…|             41.878865584|             -87.625192142|      POINT
(-87.625192…|
|040caea96573c5743…|b9a58663518c48b09…|01/01/2022 12:00:…|01/01/2022
12:15:…|        1256|     13.29|               null|                null|
76|                  null| 34.0| 0.0|  0.0|   6.0|      40.0|        Cash|
City Service|            41.980264315|            -87.913624596|    POINT
```

```
(-87.913624…|                     null|                      null|
null|
|058322b4ecd94483a…|c9867d006415cbc16…|01/01/2022 12:00:…|01/01/2022
12:00:…|           0|       0.0|               null|               null|
33|                    33| 3.25| 0.0|  0.0|   0.0|      3.25|        Cash|Taxi
Affiliation …|            41.857183858|           -87.620334624|    POINT
(-87.620334…|             41.857183858|             -87.620334624|      POINT
(-87.620334…|
|0f0c856e620e6b4df…|b21050ab3ad3d0972…|01/01/2022 12:00:…|01/01/2022
12:00:…|          33|      0.17|               null|               null|
3|                    3|63.27| 0.0|  0.0|   0.0|     63.27|        Cash|
Flash Cab|             41.96581197|            -87.655878786|    POINT
(-87.655878…|              41.96581197|             -87.655878786|      POINT
(-87.655878…|
|10de74ba327b09fc9…|86b07dc8beb256766…|01/01/2022 12:00:…|01/01/2022
12:00:…|         710|      3.12|               null|               null|
7|                    3| 11.0| 0.0|  0.0|   0.0|      11.0|        Cash|
Sun Taxi|            41.922686284|            -87.649488729|    POINT
(-87.649488…|              41.96581197|             -87.655878786|      POINT
(-87.655878…|
|1fdd5fd19aa47a9b2…|b797b5aa67c2564ed…|01/01/2022 12:00:…|01/01/2022
12:30:…|        1860|      14.4|               null|               null|
76|                  null| 37.5| 0.0|  0.0|   6.0|      43.5|        Cash| Top
Cab Affiliation|            41.980264315|            -87.913624596|    POINT
(-87.913624…|                     null|                      null|
null|
|3f11e5abdb93e75ab…|c9867d006415cbc16…|01/01/2022 12:00:…|01/01/2022
12:00:…|         300|       1.4|               null|               null|
33|                    33| 6.75| 0.0|  0.0|   0.0|      6.75|        Cash|Taxi
Affiliation …|            41.857183858|           -87.620334624|    POINT
(-87.620334…|             41.857183858|             -87.620334624|      POINT
(-87.620334…|
|43bc2cac5a899af56…|78893d83a12762723…|01/01/2022 12:00:…|01/01/2022
12:15:…|        1260|      10.4|               null|               null|
76|                  null|26.75|8.05|  0.0|   5.0|      39.8| Credit Card|Choice
Taxi Assoc…|            41.980264315|            -87.913624596|    POINT
(-87.913624…|                     null|                      null|
null|
|4c786b13744adcb24…|4ea76937237d23414…|01/01/2022 12:00:…|01/01/2022
12:15:…|         935|      4.66|               null|               null|
8|                    6|15.25| 0.0|  0.0|   0.0|     15.25|        Cash|
Sun Taxi|            41.899602111|            -87.633308037|    POINT
(-87.633308…|             41.944226601|             -87.655998182|      POINT
(-87.655998…|
|50719da0933d6056a…|d9293712880e8a69b…|01/01/2022 12:00:…|01/01/2022
12:00:…|         501|      0.65|               null|               null|
8|                    8| 6.25| 0.0|  0.0|   2.0|      8.25|        Cash|
Sun Taxi|            41.899602111|            -87.633308037|    POINT
```

```
(-87.633308…|           41.899602111|            -87.633308037|      POINT
(-87.633308…|
|52d1bd00d97eaed33…|b5e2695a2f44b9bce…|01/01/2022 12:00:…|01/01/2022
12:00:…|        598|      6.64|              null|               null|
8|                   77| 18.5| 4.0|  0.0|   1.0|      24.0| Credit Card|
Sun Taxi|           41.899602111|           -87.633308037|    POINT
(-87.633308…|             41.9867118|            -87.663416405|      POINT
(-87.663416…|
|5968a1846f875b0c0…|3c07027096c12ad3f…|01/01/2022 12:00:…|01/01/2022
12:30:…|       2254|      9.26|              null|               null|
77|                   32| 30.0| 0.0|  0.0|   0.0|      30.0|        Cash|
Sun Taxi|             41.9867118|           -87.663416405|    POINT
(-87.663416…|           41.878865584|            -87.625192142|      POINT
(-87.625192…|
|8447988f0a58c31b7…|094512e96af14b2ea…|01/01/2022 12:00:…|01/01/2022
12:15:…|       1080|       1.5|       17031081500|        17031839100|
8|                   32| 10.0| 3.4|  0.0|   1.0|      14.4| Credit Card|Taxi
Affiliation …|           41.892507781|           -87.626214906|    POINT
(-87.626214…|           41.880994471|            -87.632746489|      POINT
(-87.632746…|
|85866c8a5857f6b59…|bb4e75d3065311c33…|01/01/2022 12:00:…|01/01/2022
12:00:…|        540|       0.0|              null|               null|
8|                    7| 7.75| 2.0|  0.0|   1.5|     11.25| Credit Card|Taxi
Affiliation …|           41.899602111|           -87.633308037|    POINT
(-87.633308…|           41.922686284|            -87.649488729|      POINT
(-87.649488…|
|a64ab5107cf2b07eb…|1d8661cf286a18a51…|01/01/2022 12:00:…|01/01/2022
12:00:…|        436|       0.8|              null|               null|
null|                null|  6.0| 0.0|  0.0|   1.0|       7.0|
Cash|Chicago Independents|                    null|                   null|
null|                    null|                     null|
null|
|a9e2d462fa5af1ff6…|4cced0939feb0fece…|01/01/2022 12:00:…|01/01/2022
12:15:…|       1308|      17.9|              null|               null|
null|                null| 43.5| 9.8|  0.0|   5.0|      58.8| Credit
Card|Chicago Independents|                    null|                   null|
null|                    null|                     null|
null|
+--------------------+--------------------+--------------------+----------------
----+-----------+-----------+-------------------+--------------------+----------
-----------+----------------------+-----+----+-----+------+----------+----------
--+--------------------+------------------------+-------------------------+-----
-------------------+-------------------------+--------------------------+-------
-------------------+
only showing top 20 rows
```

taxi_spark.show() 20 DataFrame taxi_spark .

. , “Trip ID”, “Taxi ID”, “Trip Start Timestamp” .

, .

[18]:

```
#
numeric_columns = ["Trip Seconds", "Trip Miles", "Fare", "Tips", "Tolls",␣
 ↪"Extras", "Trip Total"]
for col in numeric_columns:
    taxi_spark = taxi_spark.withColumn(col, taxi_spark[col].cast(DoubleType()))
```

[19]:

```
#
int_columns = ["Pickup Community Area", "Dropoff Community Area"]
for col in int_columns:
    taxi_spark = taxi_spark.withColumn(col, taxi_spark[col].cast(IntegerType()))
```

[20]:

```
#                          TimestampType
taxi_spark = taxi_spark.withColumn("Trip Start Timestamp", to_timestamp("Trip␣
 ↪Start Timestamp", "MM/dd/yyyy HH:mm:ss"))
taxi_spark = taxi_spark.withColumn("Trip End Timestamp", to_timestamp("Trip End␣
 ↪Timestamp", "MM/dd/yyyy HH:mm:ss"))
```

[21]:

```
taxi_spark.printSchema()
```

```
root
 |-- Trip ID: string (nullable = true)
 |-- Taxi ID: string (nullable = true)
 |-- Trip Start Timestamp: timestamp (nullable = true)
 |-- Trip End Timestamp: timestamp (nullable = true)
 |-- Trip Seconds: double (nullable = true)
 |-- Trip Miles: double (nullable = true)
 |-- Pickup Census Tract: long (nullable = true)
 |-- Dropoff Census Tract: long (nullable = true)
 |-- Pickup Community Area: integer (nullable = true)
 |-- Dropoff Community Area: integer (nullable = true)
 |-- Fare: double (nullable = true)
 |-- Tips: double (nullable = true)
 |-- Tolls: double (nullable = true)
 |-- Extras: double (nullable = true)
 |-- Trip Total: double (nullable = true)
 |-- Payment Type: string (nullable = true)
 |-- Company: string (nullable = true)
 |-- Pickup Centroid Latitude: double (nullable = true)
 |-- Pickup Centroid Longitude: double (nullable = true)
 |-- Pickup Centroid Location: string (nullable = true)
 |-- Dropoff Centroid Latitude: double (nullable = true)
 |-- Dropoff Centroid Longitude: double (nullable = true)
 |-- Dropoff Centroid  Location: string (nullable = true)
```

to_timestamp() TimestampType. , ("MM/dd/yyyy HH:mm:ss") "Trip Start Timestamp" "Trip End Timestamp".

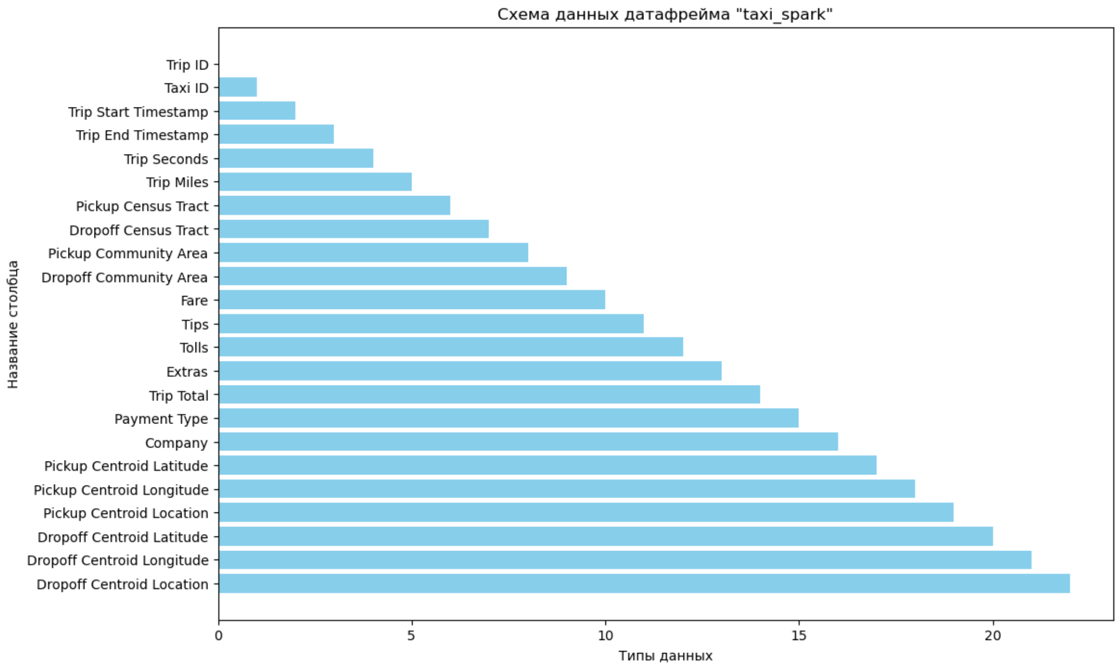
[22]:

```
#
schema = [
    ("Trip ID", "string"),
    ("Taxi ID", "string"),
    ("Trip Start Timestamp", "timestamp"),
    ("Trip End Timestamp", "timestamp"),
    ("Trip Seconds", "double"),
    ("Trip Miles", "double"),
    ("Pickup Census Tract", "long"),
    ("Dropoff Census Tract", "long"),
    ("Pickup Community Area", "integer"),
    ("Dropoff Community Area", "integer"),
    ("Fare", "double"),
    ("Tips", "double"),
    ("Tolls", "double"),
    ("Extras", "double"),
    ("Trip Total", "double"),
    ("Payment Type", "string"),
    ("Company", "string"),
    ("Pickup Centroid Latitude", "double"),
    ("Pickup Centroid Longitude", "double"),
    ("Pickup Centroid Location", "string"),
    ("Dropoff Centroid Latitude", "double"),
    ("Dropoff Centroid Longitude", "double"),
    ("Dropoff Centroid Location", "string")
]
```

[23]:

```
#
columns, data_types = zip(*schema)

#
plt.figure(figsize=(12, 8))
plt.barh(columns, range(len(columns)), color='skyblue')
plt.xlabel('      ')
plt.ylabel('         ')
plt.title('              "taxi_spark"')
plt.gca().invert_yaxis()  #           Y
plt.show()
```



[24]:
```
#                    ,
taxi_spark = taxi_spark.orderBy("Pickup Community Area", "Trip Start Timestamp")
```

[25]:
```
#                              ,
taxi_spark = taxi_spark.drop("Pickup Census Tract", "Dropoff Census Tract")
```

[26]:
```
#                           ,
aggregated_df_pickup = taxi_spark.groupBy("Pickup Community Area", F.
 ↪window("Trip Start Timestamp", "1 hour")) \
    .agg(F.count("Trip ID").alias("Total Pickup Orders"))
```

[27]:
```
#
aggregated_df_pickup = aggregated_df_pickup.withColumn("DayOfWeek", F.
 ↪dayofweek("window.start"))
aggregated_df_pickup = aggregated_df_pickup.withColumn("HourOfDay", F.
 ↪hour("window.start"))
```

[28]:
```
#
window_spec = Window.partitionBy("Pickup Community Area").orderBy("window.
 ↪start").rowsBetween(-5, 0)
aggregated_df_pickup = aggregated_df_pickup.withColumn("RollingAvgOrders", F.
 ↪avg("Total Pickup Orders").over(window_spec))

#
aggregated_df_pickup.show()
```

```
+---------------------+--------------------+-------------------+---------+------
---+------------------+
|Pickup Community Area|              window|Total Pickup
Orders|DayOfWeek|HourOfDay|  RollingAvgOrders|
+---------------------+--------------------+-------------------+---------+------
---+------------------+
|                   26|{2022-01-01 02:00…|                  2|        7|
2|               2.0|
|                   26|{2022-01-02 06:00…|                  1|        1|
6|               1.5|
|                   26|{2022-01-02 07:00…|                  1|        1|
7|1.3333333333333333|
|                   26|{2022-01-02 11:00…|                  1|        1|
11|              1.25|
|                   26|{2022-01-03 01:00…|                  2|        2|
1|               1.4|
|                   26|{2022-01-03 02:00…|                  1|        2|
2|1.3333333333333333|
|                   26|{2022-01-03 03:00…|                  1|        2|
3|1.1666666666666667|
|                   26|{2022-01-03 06:00…|                  1|        2|
6|1.1666666666666667|
|                   26|{2022-01-03 09:00…|                  2|        2|
9|1.3333333333333333|
|                   26|{2022-01-03 10:00…|                  1|        2|
10|1.3333333333333333|
|                   26|{2022-01-04 01:00…|                  1|        3|
1|1.1666666666666667|
|                   26|{2022-01-04 03:00…|                  2|        3|
3|1.3333333333333333|
|                   26|{2022-01-04 05:00…|                  1|        3|
5|1.3333333333333333|
|                   26|{2022-01-04 11:00…|                  2|        3|
11|               1.5|
|                   26|{2022-01-04 12:00…|                  1|        3|
12|1.3333333333333333|
|                   26|{2022-01-05 01:00…|                  2|        4|
1|               1.5|
|                   26|{2022-01-05 03:00…|                  2|        4|
3|1.6666666666666667|
|                   26|{2022-01-05 04:00…|                  1|        4|
4|               1.5|
|                   26|{2022-01-05 08:00…|                  1|        4|
8|               1.5|
|                   26|{2022-01-05 12:00…|                  1|        4|
12|1.3333333333333333|
+---------------------+--------------------+-------------------+---------+------
---+------------------+
only showing top 20 rows
```

[29]:
```python
aggregated_df_pickup.printSchema()
```

```
root
 |-- Pickup Community Area: integer (nullable = true)
 |-- window: struct (nullable = false)
 |    |-- start: timestamp (nullable = true)
 |    |-- end: timestamp (nullable = true)
 |-- Total Pickup Orders: long (nullable = false)
 |-- DayOfWeek: integer (nullable = true)
 |-- HourOfDay: integer (nullable = true)
 |-- RollingAvgOrders: double (nullable = true)
```

:

**Pickup Community Area ( ):** , , .

**DayOfWeek ( ):** . , , . , .

**HourOfDay ( ):** , . , .

**RollingAvgOrders ( ):** . .

**Total Pickup Orders ( ):** . .

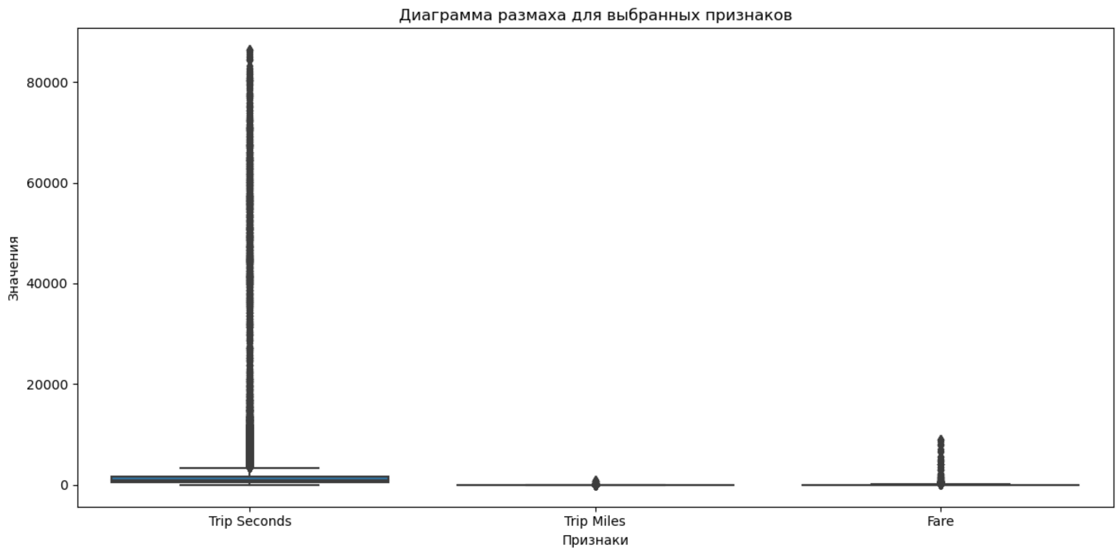
[30]:
```python
#          ,              (   , 10%)
sample_percent = 10

#                   taxi_analysis
taxi_analysis = taxi_spark.sample(withReplacement=False,␣
 ↪fraction=sample_percent / 100)
```

, .

[31]:
```python
#
selected_features = ["Trip Seconds", "Trip Miles", "Fare"]

#
plt.figure(figsize=(12, 6))
sns.boxplot(data=taxi_analysis.select(*selected_features).toPandas())
plt.title('                    ')
plt.xlabel('   ')
plt.ylabel('   ')
plt.tight_layout()
plt.show()
```



[32]:

```
#
unique_taxi_count = taxi_analysis.select(countDistinct("Taxi ID").alias("Unique Taxi IDs")).collect()[0][0]
unique_taxi_count
```

[32]: 3234

[33]:

```
taxi_analysis.groupBy('Taxi ID').agg(
    F.count('Taxi ID').alias('count_trip'),
    F.sum('Trip Seconds').alias('sum_seconds'),
    F.round(F.sum('Trip Miles')).alias('sum_miles'),
    F.round(F.sum('Trip Total')).alias('trip_total'),
    F.avg('Fare').alias('avg_fare'),
    F.sum('Tips').alias('total_tips')
).show(10)
```

```
+--------------------+----------+-----------+---------+----------+--------------
----+------------------+
|             Taxi ID|count_trip|sum_seconds|sum_miles|trip_total|
avg_fare|        total_tips|
+--------------------+----------+-----------+---------+----------+--------------
----+------------------+
|8314611044ff50100…|       146|   258603.0|   1954.0|    6569.0|
34.98458904109589| 806.9200000000002|
|4972764cee12598f4…|       554|   714303.0|   3725.0|
14372.0|21.929205776173283|1402.3400000000001|
|d2c2d4128d6597a3b…|       579|   643961.0|   2873.0|
11771.0|18.401001727115716| 806.8899999999999|
|f6d1b6c930d62f6d8…|       541|   605191.0|   2856.0|
11173.0|16.635619223659887|           1520.86|
|b5bf5d282fa4191c6…|       544|   525960.0|      0.0|   12906.0|
20.59862132352941|1423.0199999999998|
|26edb3e8696634e74…|       213|   273900.0|     91.0|
6229.0|22.435446009389672| 821.5199999999999|
|074ebefb524b3c9c3…|       289|   485943.0|   3576.0|   12611.0|
32.66335640138408|1636.4899999999998|
|1e4ba7f6a2c79ac22…|       355|   428667.0|   2864.0|
11086.0|23.553464788732395|1628.5299999999997|
|da1fa60939f1104bf…|       177|   231264.0|   1330.0|
5217.0|22.496214689265535|            660.78|
|8d9a2218e0a2c8ae9…|       546|   480629.0|   1969.0|    9255.0|
13.85551282051282|1001.7099999999999|
+------------------+----------+-----------+---------+----------+-------------
----+------------------+
only showing top 10 rows
```

[34]:

```python
#
pickup_areas = taxi_analysis.select("Pickup Community Area").distinct()
dropoff_areas = taxi_analysis.select("Dropoff Community Area").distinct()

#                   Python
pickup_areas_list = [row["Pickup Community Area"] for row in pickup_areas.
 ↪collect() if row["Pickup Community Area"] is not None]
dropoff_areas_list = [row["Dropoff Community Area"] for row in dropoff_areas.
 ↪collect() if row["Dropoff Community Area"] is not None]
```

[35]:

```python
#     ,
if len(pickup_areas_list) == len(dropoff_areas_list):
    print("                                                 :",␣
 ↪len(pickup_areas_list))
else:
    print("                                                   .")
```

```
                                             :
77
```

[36]:

```python
#
taxi_analysis = taxi_analysis.filter(
    F.col("Pickup Centroid Latitude").isNotNull() &
    F.col("Pickup Centroid Longitude").isNotNull() &
    F.col("Dropoff Centroid Latitude").isNotNull() &
    F.col("Dropoff Centroid Longitude").isNotNull()
)
```

[37]:
```
#
m = folium.Map(location=[41.8781, -87.6298], zoom_start=10)  #
```

[38]:
```
#
pickup_heatmap_data = taxi_analysis.select("Pickup Centroid Latitude", "Pickup Centroid Longitude").collect()
pickup_heatmap = HeatMap(pickup_heatmap_data, radius=15)
pickup_heatmap.add_to(m)
```

[38]:
```
<folium.plugins.heat_map.HeatMap at 0x7fba549f2f90>
```

[39]:
```
#
dropoff_heatmap_data = taxi_analysis.select("Dropoff Centroid Latitude", "Dropoff Centroid Longitude").collect()
dropoff_heatmap = HeatMap(dropoff_heatmap_data, radius=15)
dropoff_heatmap.add_to(m)
```

[39]:
```
<folium.plugins.heat_map.HeatMap at 0x7fb9bdb8b650>
```

[40]:
```
#
m
```

[40]:
```
<folium.folium.Map at 0x7fba549a4910>
```

, , . ,
, , ,
. .

[41]:
```
#                    'Company'
company_counts = taxi_analysis.groupBy('Company').count()

#                                             ,
most_frequent_companies = company_counts.orderBy(F.col('count').desc())

#         N     ,
most_frequent_companies.show(10)
```

```
+--------------------+------+
|             Company| count|
+--------------------+------+
|Taxi Affiliation …|185741|
|           Flash Cab|184808|
|            Sun Taxi| 97645|
|        City Service| 87199|
|Taxicab Insurance…| 56075|
|Chicago Independents| 46470|
|    Medallion Leasin| 34352|
|          Globe Taxi| 32831|
|Taxicab Insurance…| 28960|
|         5 Star Taxi| 27423|
+--------------------+-------+
only showing top 10 rows
```

, : “Taxi Affiliation Services” “Flash Cab” 180 000 184 000 . .

[42]:
```
from pyspark.sql.functions import col

#            ,
trips_with_tips = taxi_analysis.filter(col("Tips") > 0)
total_trips_with_tips = trips_with_tips.count()

#
total_trips = taxi_analysis.count()

total_trips_with_tips, total_trips
```

[42]:
```
(452123, 885123)
```

, 452,102 887,003 ( 51%) . , . , , , , .

[43]:
```
#
payment_counts = taxi_analysis.groupBy('Payment Type').count().orderBy('count',␣
 ↪ascending=False)

#
print("                         :")
payment_counts.show()
```

```
                          :
+------------+------+
|Payment Type| count|
+------------+------+
| Credit Card|330962|
|        Cash|274211|
|      Mobile|144270|
|      Prcard| 89932|
|     Unknown| 44882|
|   No Charge|   331|
|     Dispute|   292|
+------------+------+
```

:

- (Credit Card): 332,574 .
- (Cash): 274,435 .
- (Mobile): 144,505 .
- (Prcard): 89,248 .
- (Unknown): 45,598 .
- (Dispute): 303 .
- (No Charge): 301 .

(Credit Card) (Cash). , 332,000 , 274,000 . , (Mobile), (Prcard) , .

[44]:

```
#
payment_sums = taxi_analysis.groupBy('Payment Type').agg(F.sum('Trip Total').alias('Total Amount')).orderBy('Total Amount', ascending=False)

print("                    :")
payment_sums.show()
```

```
                :
+------------+--------------------+
|Payment Type|        Total Amount|
+------------+--------------------+
| Credit Card|1.0730066029999923E7|
|        Cash|   4596717.900000009|
|      Mobile|  2650026.1300000097|
|      Prcard|  2241006.0100000054|
|     Unknown|          1028289.56|
|     Dispute|             9749.57|
|   No Charge|             7987.97|
+------------+--------------------+
```

[45]:

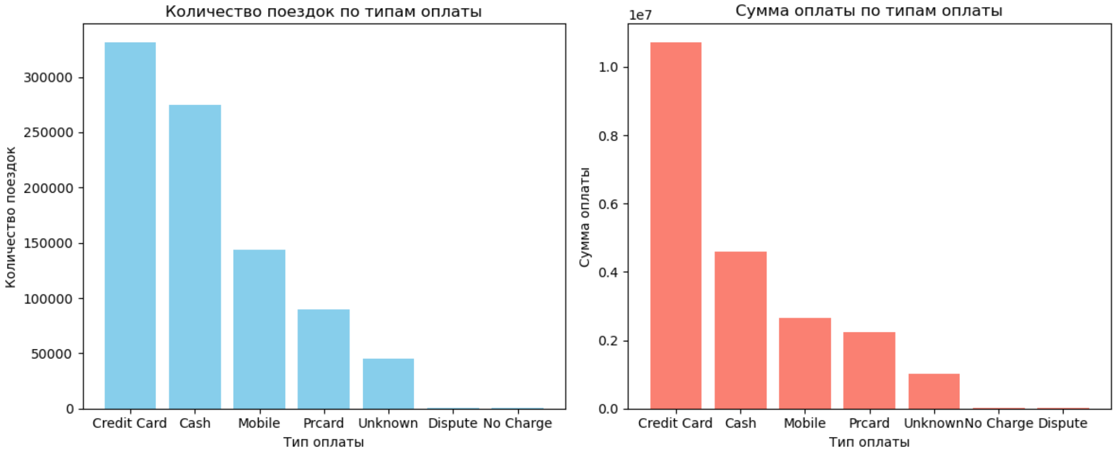
```
#
payment_counts_pd = payment_counts.toPandas()
payment_sums_pd = payment_sums.toPandas()

plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
plt.bar(payment_counts_pd['Payment Type'], payment_counts_pd['count'], color='skyblue')
plt.title('                    ')
plt.xlabel('       ')
plt.ylabel('           ')

plt.subplot(1, 2, 2)
plt.bar(payment_sums_pd['Payment Type'], payment_sums_pd['Total Amount'],␣
  ↪color='salmon')
plt.title('                 ')
plt.xlabel('       ')
plt.ylabel('        ')

plt.tight_layout()
plt.show()
```



, :

1. **(Credit Card)** . $10,774,000, , .

2. **(Cash)** $4,607,944. .

3. **(Mobile)** , $2,646,907. .

4. **(Prcard)** , $2,250,491. .

5. **(Unknown)** $1,022,605.89. , “Unknown” .

6. **(No Charge)** **(Dispute)** , $7,798.86 $6,084.60 .

, ,
.

, $7,798.86. , . ,
, , - .

,
.

[46]:
```
from pyspark.sql.functions import col

#                                $7,798.86
free_trips_with_amount = taxi_analysis.filter((col("Payment Type") == "No␣
 ↪Charge") & (col("Trip Total") == 7798.86))

#
free_trips_with_amount.show()
```

```
+-------+-------+--------------------+------------------+------------+----------
+---------------------+----------------------+----+----+-----+------+----------
+------------+-------+------------------------+-------------------------+-------
-----------------+-------------------------+--------------------------+---------
-----------------+
|Trip ID|Taxi ID|Trip Start Timestamp|Trip End Timestamp|Trip Seconds|Trip
Miles|Pickup Community Area|Dropoff Community Area|Fare|Tips|Tolls|Extras|Trip
Total|Payment Type|Company|Pickup Centroid Latitude|Pickup Centroid
Longitude|Pickup Centroid Location|Dropoff Centroid Latitude|Dropoff Centroid
Longitude|Dropoff Centroid  Location|
+-------+-------+--------------------+------------------+------------+----------
+---------------------+----------------------+----+----+-----+------+----------
+------------+-------+------------------------+-------------------------+-------
-----------------+-------------------------+--------------------------+---------
-----------------+
+-------+-------+--------------------+------------------+------------+----------
+---------------------+----------------------+----+----+-----+------+----------
+------------+-------+------------------------+-------------------------+-------
-----------------+-------------------------+--------------------------+---------
-----------------+
```

, $7,798.86. ,
, .

[47]:
```
#         ,    "Payment Type"     "No Charge"
no_charge_trips = taxi_analysis.filter(col("Payment Type") == "No Charge")

#
count_no_charge_trips = no_charge_trips.count()
```

```
#
total_tips_in_no_charge_trips = no_charge_trips.agg({"Tips": "sum"}).
 ↪collect()[0][0]

count_no_charge_trips, total_tips_in_no_charge_trips
```

[47]:
```
(284, 76.80000000000001)
```

, :

- “No Charge” 285.
- $63.19.

, 285 ( “No Charge”), $63.19. , - , - .

[48]:
```
#                     "Pickup Community Area"
taxi_spark = taxi_spark.join(aggregated_df_pickup,
    on="Pickup Community Area",
   how="left")

#
taxi_spark.printSchema()
```

```
root
 |-- Pickup Community Area: integer (nullable = true)
 |-- Trip ID: string (nullable = true)
 |-- Taxi ID: string (nullable = true)
 |-- Trip Start Timestamp: timestamp (nullable = true)
 |-- Trip End Timestamp: timestamp (nullable = true)
 |-- Trip Seconds: double (nullable = true)
 |-- Trip Miles: double (nullable = true)
 |-- Dropoff Community Area: integer (nullable = true)
 |-- Fare: double (nullable = true)
 |-- Tips: double (nullable = true)
 |-- Tolls: double (nullable = true)
 |-- Extras: double (nullable = true)
 |-- Trip Total: double (nullable = true)
 |-- Payment Type: string (nullable = true)
 |-- Company: string (nullable = true)
 |-- Pickup Centroid Latitude: double (nullable = true)
 |-- Pickup Centroid Longitude: double (nullable = true)
 |-- Pickup Centroid Location: string (nullable = true)
 |-- Dropoff Centroid Latitude: double (nullable = true)
 |-- Dropoff Centroid Longitude: double (nullable = true)
 |-- Dropoff Centroid  Location: string (nullable = true)
 |-- window: struct (nullable = true)
 |    |-- start: timestamp (nullable = true)
 |    |-- end: timestamp (nullable = true)
 |-- Total Pickup Orders: long (nullable = true)
 |-- DayOfWeek: integer (nullable = true)
 |-- HourOfDay: integer (nullable = true)
 |-- RollingAvgOrders: double (nullable = true)
```

DataFrame taxi_spark aggregated_df_pickup “Pickup

# 2

[49]:
```python
#
for column_name in taxi_spark.columns:
    new_column_name = column_name.lower().replace(' ', '_')
    taxi_spark = taxi_spark.withColumnRenamed(column_name, new_column_name)
```

[50]:
```python
taxi_spark.printSchema()
```

```
root
 |-- pickup_community_area: integer (nullable = true)
 |-- trip_id: string (nullable = true)
 |-- taxi_id: string (nullable = true)
 |-- trip_start_timestamp: timestamp (nullable = true)
 |-- trip_end_timestamp: timestamp (nullable = true)
 |-- trip_seconds: double (nullable = true)
 |-- trip_miles: double (nullable = true)
 |-- dropoff_community_area: integer (nullable = true)
 |-- fare: double (nullable = true)
 |-- tips: double (nullable = true)
 |-- tolls: double (nullable = true)
 |-- extras: double (nullable = true)
 |-- trip_total: double (nullable = true)
 |-- payment_type: string (nullable = true)
 |-- company: string (nullable = true)
 |-- pickup_centroid_latitude: double (nullable = true)
 |-- pickup_centroid_longitude: double (nullable = true)
 |-- pickup_centroid_location: string (nullable = true)
 |-- dropoff_centroid_latitude: double (nullable = true)
 |-- dropoff_centroid_longitude: double (nullable = true)
 |-- dropoff_centroid__location: string (nullable = true)
 |-- window: struct (nullable = true)
 |    |-- start: timestamp (nullable = true)
 |    |-- end: timestamp (nullable = true)
 |-- total_pickup_orders: long (nullable = true)
 |-- dayofweek: integer (nullable = true)
 |-- hourofday: integer (nullable = true)
 |-- rollingavgorders: double (nullable = true)
```

[51]:

```
#
df_selection = taxi_spark.select(
    'pickup_community_area', 'trip_seconds', 'trip_miles', 'fare', 'tips',␣
 ↪'extras', 'taxi_id',
    'payment_type', 'company', 'total_pickup_orders', 'dayofweek', 'hourofday',␣
 ↪'rollingavgorders', 'trip_start_timestamp')
```

[52]:

```
#
def make_features(data, max_lag, rolling_mean_size):
    lag_window = Window.orderBy('trip_start_timestamp')
    mean_window = Window.orderBy('trip_start_timestamp').rowsBetween(-1 -␣
 ↪rolling_mean_size, -1)

    #             df_selection
    data = (data.select(*df_selection.columns)
            .withColumn('day_of_month', F.dayofmonth('trip_start_timestamp'))
            .withColumn('day_of_week', F.dayofweek(F.
 ↪col('trip_start_timestamp')))
            .withColumn('hour_of_day', F.hour('trip_start_timestamp'))
            .withColumn('trip_seconds_avg', F.avg('trip_seconds').
 ↪over(mean_window))
            .withColumn('trip_miles_avg', F.avg('trip_miles').over(mean_window))
            .withColumn('fare_avg', F.avg('fare').over(mean_window))
            .withColumn('tips_avg', F.avg('tips').over(mean_window))
            .withColumn('extras_avg', F.avg('extras').over(mean_window))
            .withColumn('taxi_distinct', F.count('taxi_id').over(mean_window))
            .withColumn('payment_type_distinct', F.count('payment_type').
 ↪over(mean_window))
            .withColumn('company_distinct', F.count('company').
 ↪over(mean_window))
           )

    for lag in range(1, max_lag + 1):
        #                              df_selection
        #     ,    'fare'            F.lag('fare', lag).over(lag_window)
        data = data.withColumn('lag_{}'.format(lag), F.lag('fare', lag).
 ↪over(lag_window))

    data = data.dropna()
    return data
```

## 3

[79]:

```
#
target = 'total_pickup_orders'
df_selection = df_selection.withColumnRenamed(target, "label")
```

[80]:

```
#
split_weights = [0.6, 0.2, 0.2]

#                  (seed)
seed = 12345

#         randomSplit                     seed
split_data = df_selection.randomSplit(split_weights, seed=seed)

#                  ,
train_data = split_data[0]
valid_data = split_data[1]
test_data = split_data[2]
```

[81]:

```
#
max_lag = 24   #
rolling_mean_size = 24   #

#                             ,
train_data = make_features(df_selection, max_lag, rolling_mean_size)
valid_data = make_features(df_selection, max_lag, rolling_mean_size)
test_data = make_features(df_selection, max_lag, rolling_mean_size)
```

[82]:

```
train_data.printSchema()
```

```
root
 |-- pickup_community_area: integer (nullable = true)
 |-- trip_seconds: double (nullable = true)
 |-- trip_miles: double (nullable = true)
 |-- fare: double (nullable = true)
 |-- tips: double (nullable = true)
 |-- extras: double (nullable = true)
 |-- taxi_id: string (nullable = true)
 |-- payment_type: string (nullable = true)
 |-- company: string (nullable = true)
 |-- label: long (nullable = true)
 |-- dayofweek: integer (nullable = true)
 |-- hourofday: integer (nullable = true)
 |-- rollingavgorders: double (nullable = true)
 |-- trip_start_timestamp: timestamp (nullable = true)
 |-- day_of_month: integer (nullable = true)
 |-- day_of_week: integer (nullable = true)
 |-- hour_of_day: integer (nullable = true)
 |-- trip_seconds_avg: double (nullable = true)
 |-- trip_miles_avg: double (nullable = true)
 |-- fare_avg: double (nullable = true)
 |-- tips_avg: double (nullable = true)
 |-- extras_avg: double (nullable = true)
 |-- taxi_distinct: long (nullable = false)
 |-- payment_type_distinct: long (nullable = false)
 |-- company_distinct: long (nullable = false)
 |-- lag_1: double (nullable = true)
 |-- lag_2: double (nullable = true)
 |-- lag_3: double (nullable = true)
 |-- lag_4: double (nullable = true)
 |-- lag_5: double (nullable = true)
 |-- lag_6: double (nullable = true)
 |-- lag_7: double (nullable = true)
 |-- lag_8: double (nullable = true)
 |-- lag_9: double (nullable = true)
 |-- lag_10: double (nullable = true)
 |-- lag_11: double (nullable = true)
 |-- lag_12: double (nullable = true)
 |-- lag_13: double (nullable = true)
 |-- lag_14: double (nullable = true)
 |-- lag_15: double (nullable = true)
 |-- lag_16: double (nullable = true)
 |-- lag_17: double (nullable = true)
 |-- lag_18: double (nullable = true)
 |-- lag_19: double (nullable = true)
 |-- lag_20: double (nullable = true)
 |-- lag_21: double (nullable = true)
 |-- lag_22: double (nullable = true)
 |-- lag_23: double (nullable = true)
 |-- lag_24: double (nullable = true)
```

[83]:

```
#
cat_features = ['dayofweek', 'hourofday', 'day_of_month', 'day_of_week',␣
 ↪'hour_of_day']

num_features = [
     'trip_seconds', 'trip_miles', 'fare', 'tips', 'extras','rollingavgorders',␣
 ↪'trip_seconds_avg', 'trip_miles_avg', 'fare_avg', 'tips_avg', 'extras_avg',
     'taxi_distinct', 'payment_type_distinct', 'company_distinct',
     'lag_1', 'lag_2', 'lag_3', 'lag_4', 'lag_5', 'lag_6', 'lag_7', 'lag_8',␣
 ↪'lag_9', 'lag_10',
     'lag_11', 'lag_12', 'lag_13', 'lag_14', 'lag_15', 'lag_16', 'lag_17',␣
 ↪'lag_18', 'lag_19', 'lag_20',
    'lag_21', 'lag_22', 'lag_23', 'lag_24']
```

[84]:
```
encoder = OneHotEncoder(inputCols=cat_features, outputCols=[c + '_ohe' for c in␣
  ↪cat_features])
```

[85]:
```
num_assembler = VectorAssembler(inputCols=num_features,␣
  ↪outputCol='num_features')
```

[86]:
```
scaler = StandardScaler(inputCol='num_features',␣
  ↪outputCol='num_features_scaled')
```

[87]:
```
assembler_lr = VectorAssembler(inputCols=encoder.getOutputCols() +␣
  ↪['num_features_scaled'], outputCol='features')

assembler = VectorAssembler(inputCols=(cat_features + num_features),␣
  ↪outputCol='features')
```

[88]:
```
#
random_forest_model = RandomForestRegressor(featuresCol='features',␣
  ↪labelCol='label')
decision_tree_model = DecisionTreeRegressor(featuresCol='features',␣
  ↪labelCol='label')
linear_regression_model = LinearRegression(featuresCol='num_features_scaled',␣
  ↪labelCol='label')
```

[89]:
```
#
evaluator = RegressionEvaluator(predictionCol='prediction', labelCol='label',␣
  ↪metricName='rmse')
```

[90]:
```
#         Random Forest
pipeline_random_forest = Pipeline(stages=[assembler, random_forest_model])

#         Decision Tree
pipeline_decision_tree = Pipeline(stages=[assembler, decision_tree_model])

#         Linear Regression
pipeline_linear_regression = Pipeline(stages=[encoder, num_assembler, scaler,␣
  ↪assembler_lr, linear_regression_model])
```

[ ]:
```
#
rf_model = pipeline_random_forest.fit(train_data)
```

[ ]:
```
dt_model = pipeline_decision_tree.fit(train_data)
```

[ ]:
```
lr_model = pipeline_linear_regression.fit(train_data)
```

[ ]:

```
#
rf_predictions_valid = rf_model.transform(valid_data)
```

[ ]:

```
dt_predictions_valid = dt_model.transform(valid_data)
```

[ ]:

```
lr_predictions_valid = lr_model.transform(valid_data)
```

[ ]:

```
#      RMSE
rf_rmse_valid = evaluator.evaluate(rf_predictions_valid)
dt_rmse_valid = evaluator.evaluate(dt_predictions_valid)
lr_rmse_valid = evaluator.evaluate(lr_predictions_valid)
```

[ ]:

```
print("RMSE   Random Forest              :", rf_rmse_valid)
print("RMSE   Decision Tree              :", dt_rmse_valid)
print("RMSE   Linear Regression              :", lr_rmse_valid)
```

### 3.0.1

[ ]:

```
rf_predictions_test = rf_model.transform(test_data)
```