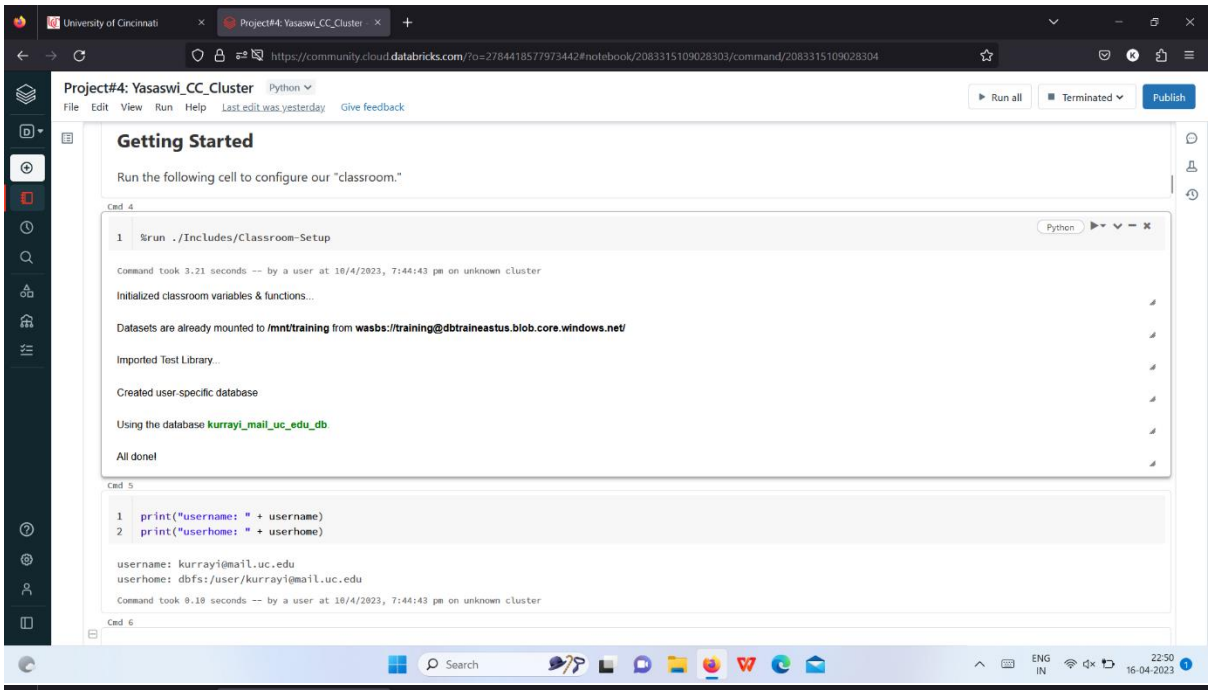
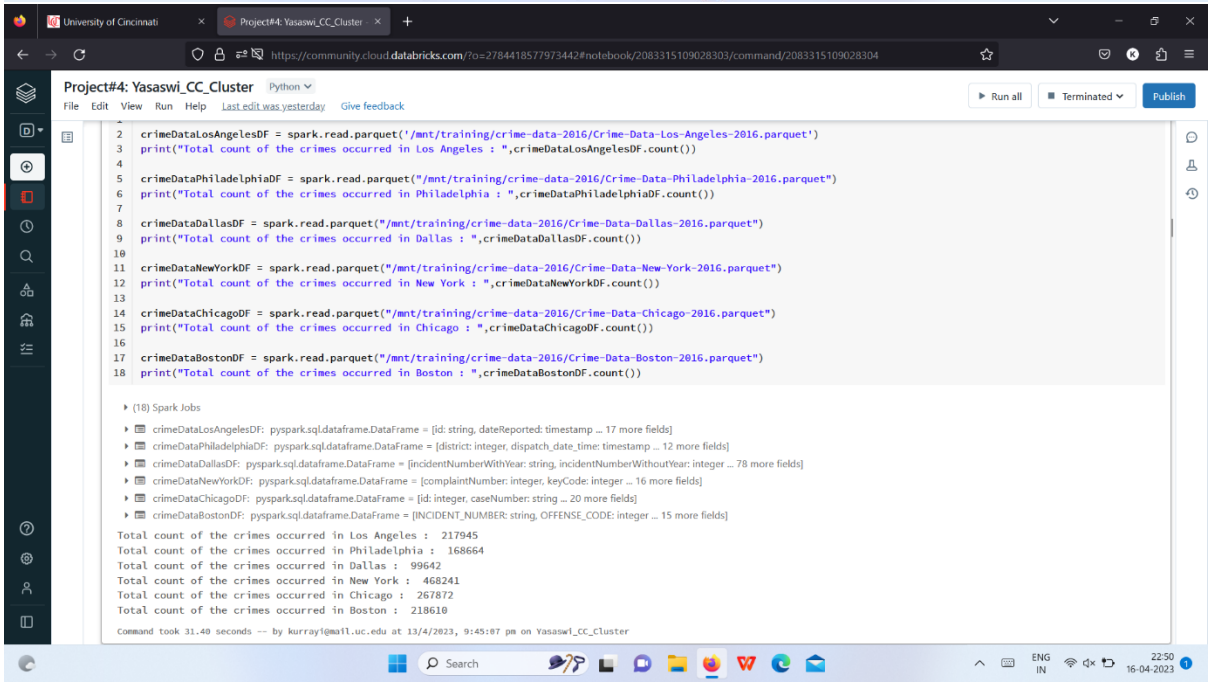
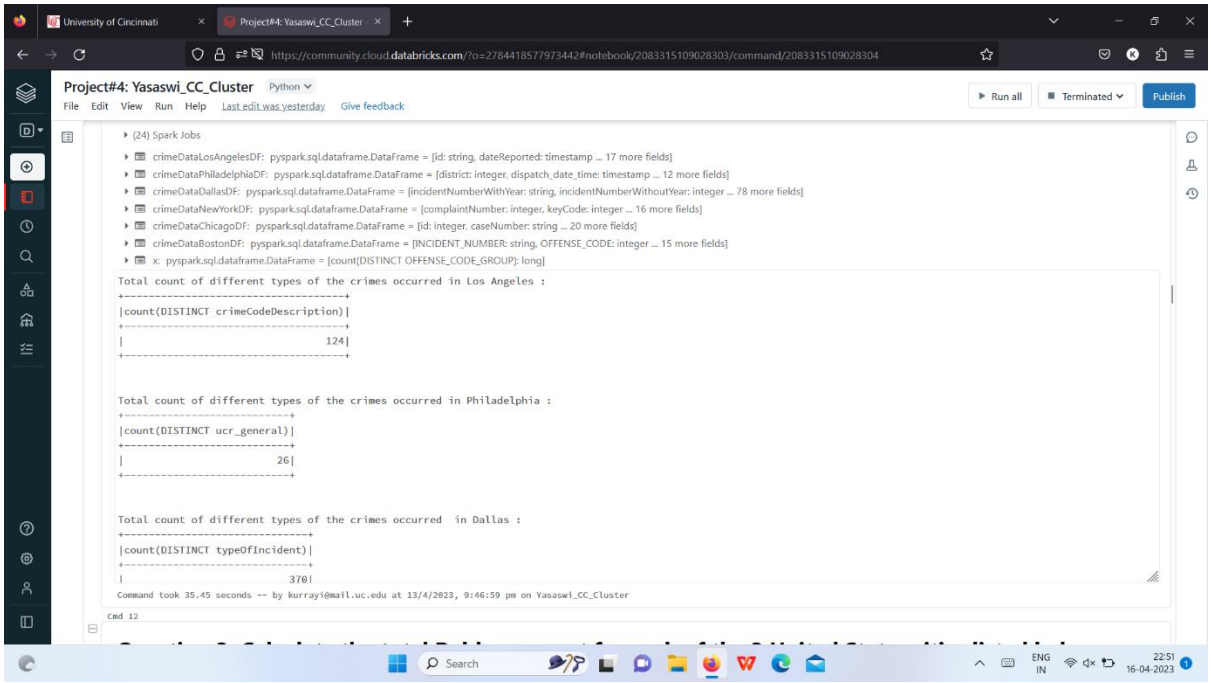
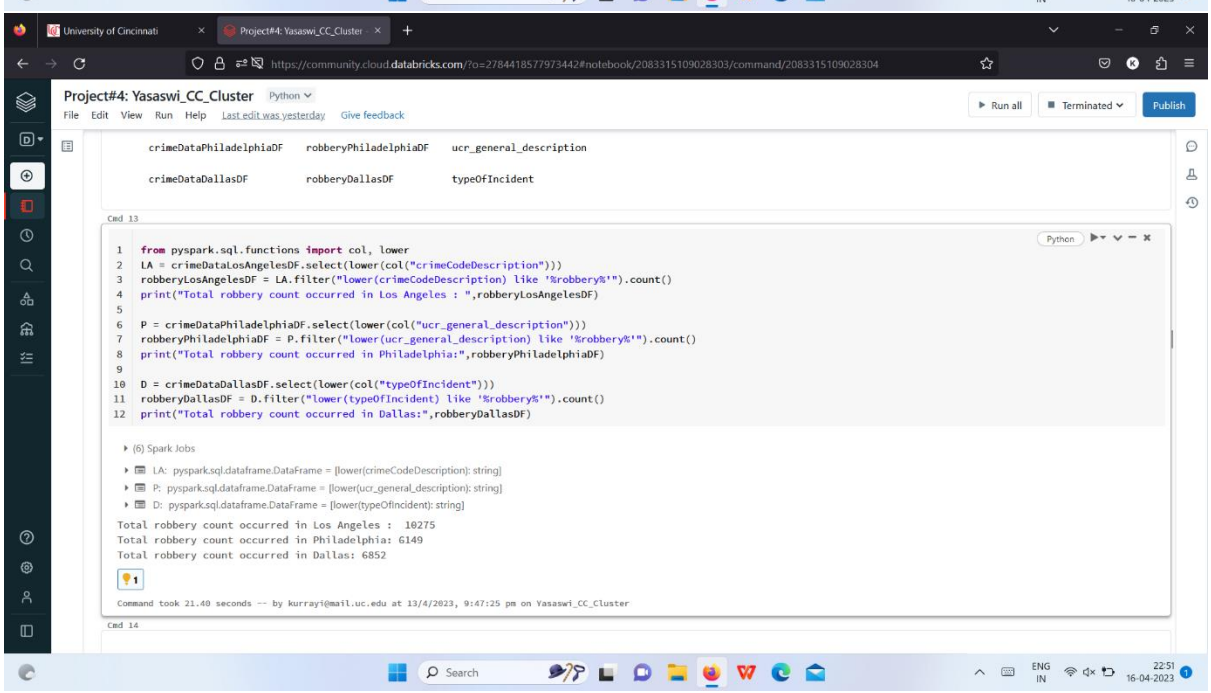
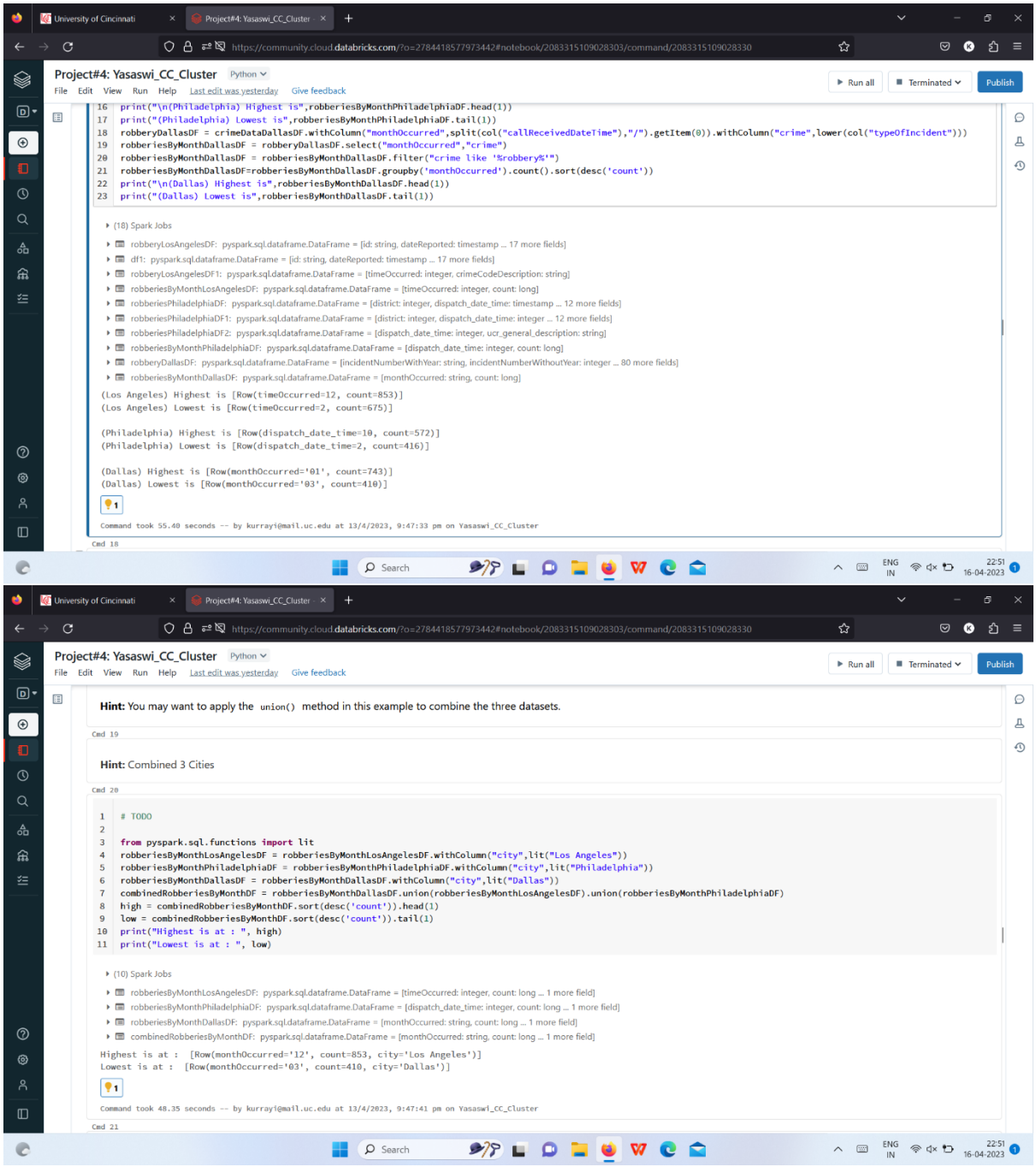
Cloud Computing



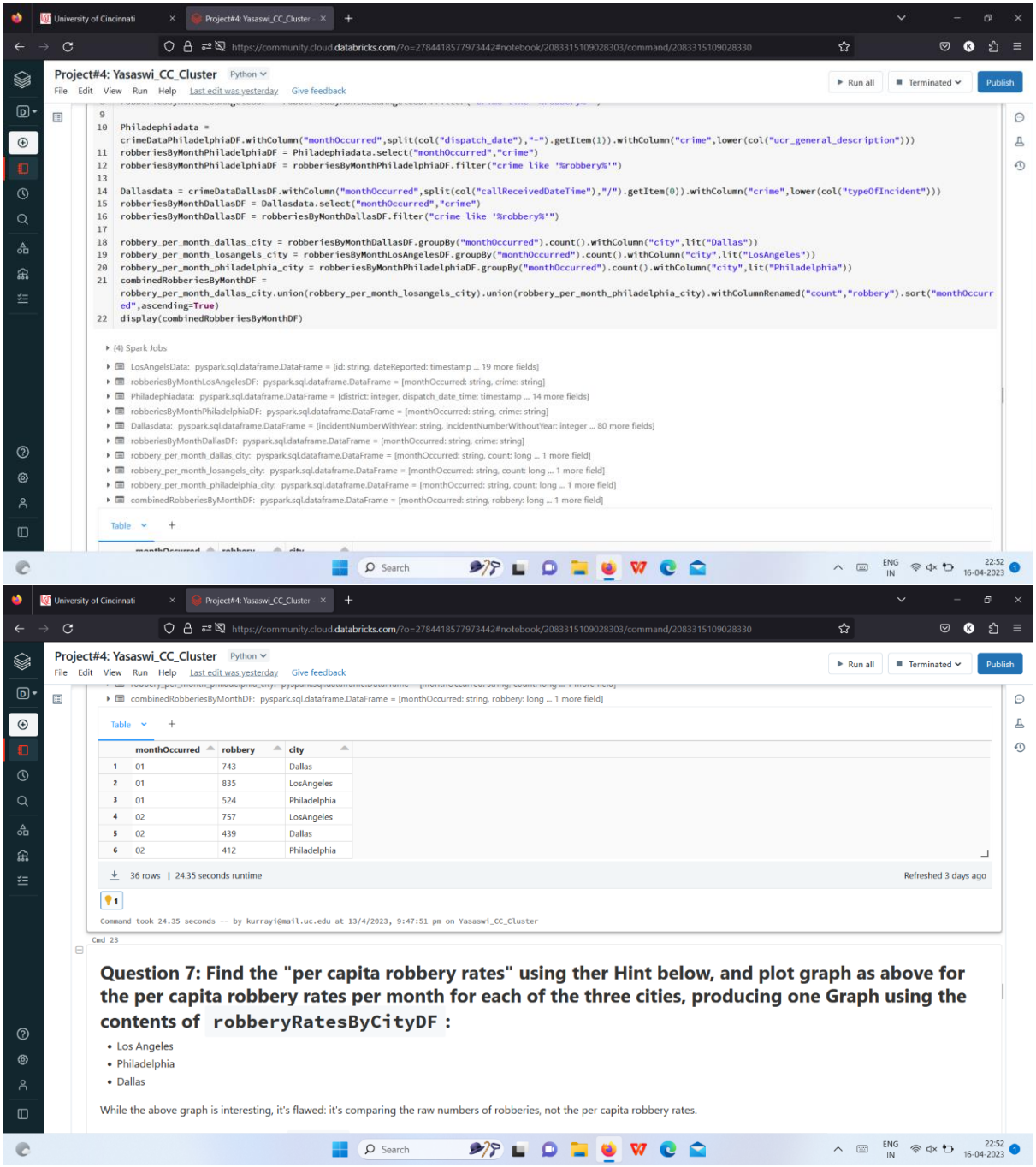



University of Cincinnati
Project#4: Yasaswi_CC_Cluster
https://community.cloud.databricks.com/?o=2784418577973442#notebook/2083315109028303/command/2083315109028304
Project#4: Yasaswi_CC_Cluster
Python
File Edit View Run Help Last edit was yesterday Give feedback
Run all
Terminated
Publish
(24) Spark Jobs
crimeDataLosAngelesDF: pyspark.sql.dataframe.DataFrame = [id: string, dateReported: timestamp ... 17 more fields]
crimeDataPhiladelphiaDF: pyspark.sql.dataframe.DataFrame = [district: integer, dispatch_date_time: timestamp ... 12 more fields]
crimeDataDallasDF: pyspark.sql.dataframe.DataFrame = [incidentNumberWithYear: string, incidentNumberWithoutYear: integer ... 78 more fields]
crimeDataNewYorkDF: pyspark.sql.dataframe.DataFrame = [complaintNumber: integer, keyCode: integer ... 16 more fields]
crimeDataChicagoDF: pyspark.sql.dataframe.DataFrame = [id: integer, caseNumber: string ... 20 more fields]
crimeDataBostonDF: pyspark.sql.dataframe.DataFrame = [INCIDENT_NUMBER: string, OFFENSE_CODE: integer ... 15 more fields]
x: pyspark.sql.dataframe.DataFrame = [count(DISTINCT OFFENSE_CODE_GROUP): long]
Total count of different types of the crimes occurred in Los Angeles :
+---------------------------------------+
|count(DISTINCT crimeCodeDescription)|
+---------------------------------------+
| 124|
+---------------------------------------+
Total count of different types of the crimes occurred in Philadelphia :
+----------------------------+
|count(DISTINCT ucr_general)|
+----------------------------+
| 26|
+----------------------------+
Total count of different types of the crimes occurred in Dallas :
+-------------------------------+
|count(DISTINCT typeOfIncident)|
+-------------------------------+
| 370|
+-------------------------------+
Command took 35.45 seconds -- by kurrayi@mail.uc.edu at 13/4/2023, 9:46:59 pm on Yasaswi_CC_Cluster
Cmd 12
Search
ENG IN
22:51 16-04-2023

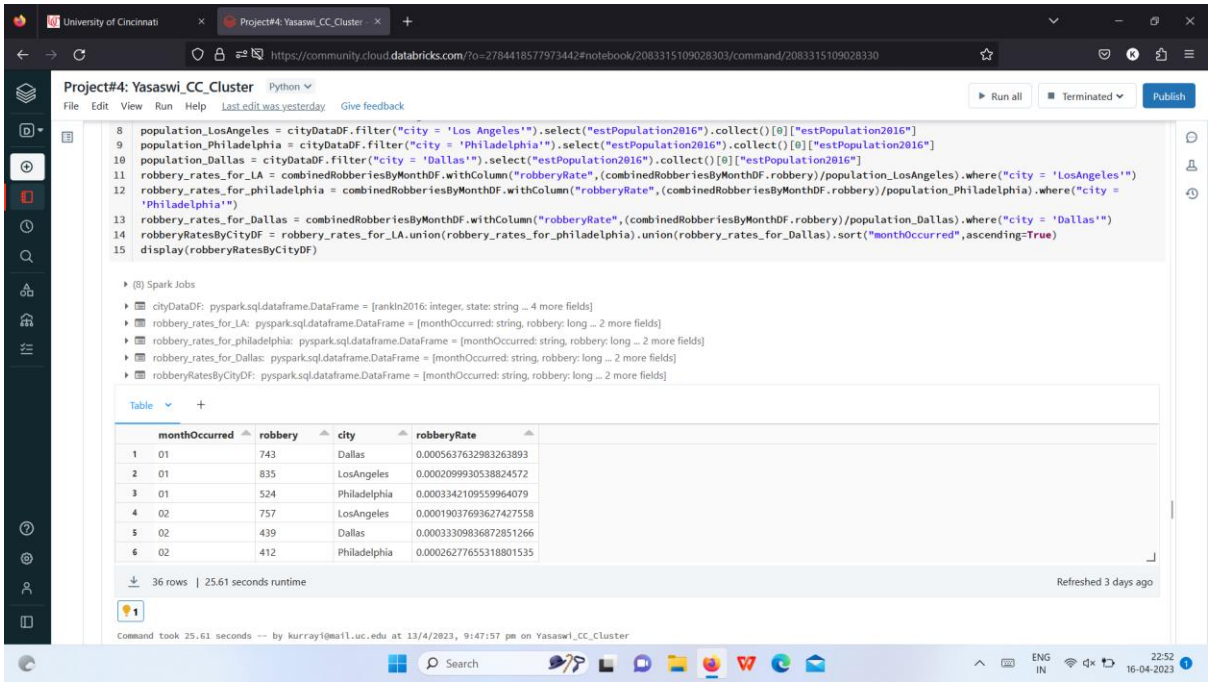

University of Cincinnati
Project#4: Yasaswi_CC_Cluster
https://community.cloud.databricks.com/?o=2784418577973442#notebook/2083315109028303/command/2083315109028304
Project#4: Yasaswi_CC_Cluster
Python
File Edit View Run Help Last edit was yesterday Give feedback
Run all
Terminated
Publish
crimeDataPhiladelphiaDF robberyPhiladelphiaDF ucr_general_description
crimeDataDallasDF robberyDallasDF typeOfIncident
Cmd 13
Python
1 from pyspark.sql.functions import col, lower
2 LA = crimeDataLosAngelesDF.select(lower(col("crimeCodeDescription")))
3 robberyLosAngelesDF = LA.filter("lower(crimeCodeDescription) like '%robbery%'").count()
4 print("Total robbery count occurred in Los Angeles : ",robberyLosAngelesDF)
5
6 P = crimeDataPhiladelphiaDF.select(lower(col("ucr_general_description")))
7 robberyPhiladelphiaDF = P.filter("lower(ucr_general_description) like '%robbery%'").count()
8 print("Total robbery count occurred in Philadelphia:",robberyPhiladelphiaDF)
9
10 D = crimeDataDallasDF.select(lower(col("typeOfIncident")))
11 robberyDallasDF = D.filter("lower(typeOfIncident) like '%robbery%'").count()
12 print("Total robbery count occurred in Dallas:",robberyDallasDF)
(6) Spark Jobs
LA: pyspark.sql.dataframe.DataFrame = [lower(crimeCodeDescription): string]
P: pyspark.sql.dataframe.DataFrame = [lower(ucr_general_description): string]
D: pyspark.sql.dataframe.DataFrame = [lower(typeOfIncident): string]
Total robbery count occurred in Los Angeles : 10275
Total robbery count occurred in Philadelphia: 6149
Total robbery count occurred in Dallas: 6852
Command took 21.40 seconds -- by kurrayi@mail.uc.edu at 13/4/2023, 9:47:25 pm on Yasaswi_CC_Cluster
Cmd 14
Search
ENG IN
22:51 16-04-2023

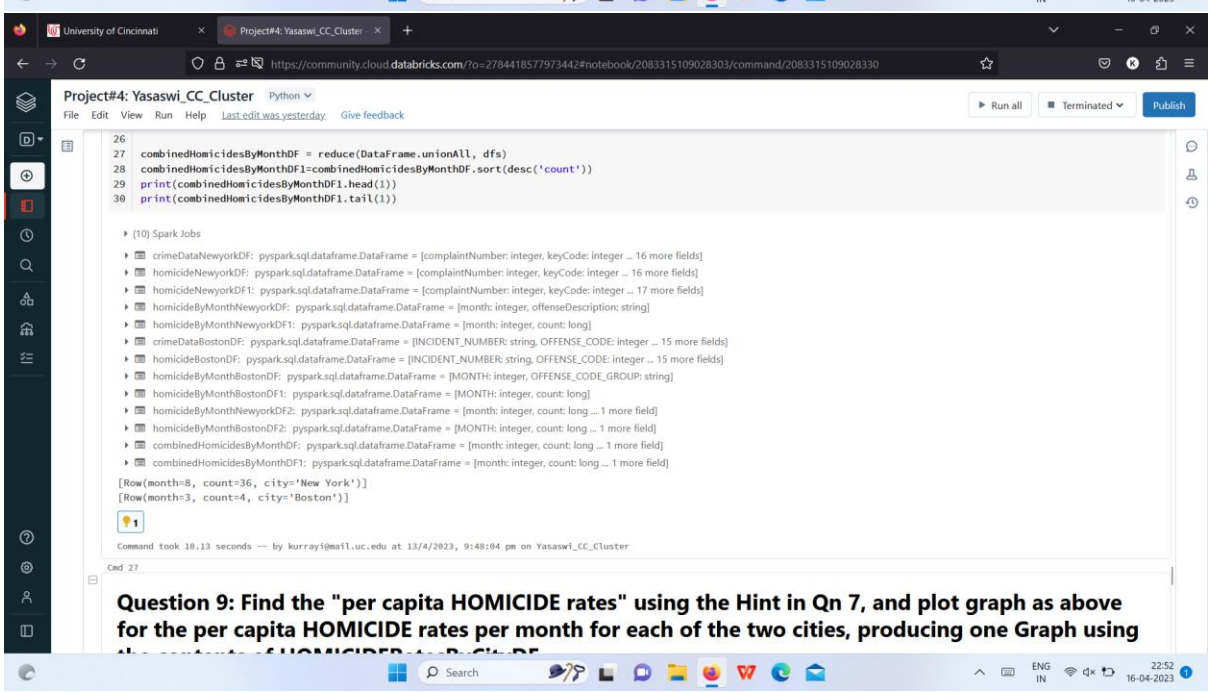

University of Cincinnati
Project#4: Yasaswi_CC_Cluster
https://community.cloud.databricks.com/?o=2784418577973442#notebook/2083315109028303/command/2083315109028330
Project#4: Yasaswi_CC_Cluster
Python
File Edit View Run Help Last edit was yesterday Give feedback
Run all
Terminated
Publish
print("\n(Philadelphia) Highest is",robberiesByMonthPhiladelphiaDF.head(1))
print("(Philadelphia) Lowest is",robberiesByMonthPhiladelphiaDF.tail(1))
robberyDallasDF = crimeDataDallasDF.withColumn("monthOccurred",split(col("callReceivedDateTime"),"/").getItem(0)).withColumn("crime",lower(col("typeOfIncident")))
robberiesByMonthDallasDF = robberyDallasDF.select("monthOccurred","crime")
robberiesByMonthDallasDF = robberiesByMonthDallasDF.filter("crime like '%robbery%'")
robberiesByMonthDallasDF=robberiesByMonthDallasDF.groupby('monthOccurred').count().sort(desc('count'))
print("\n(Dallas) Highest is",robberiesByMonthDallasDF.head(1))
print("(Dallas) Lowest is",robberiesByMonthDallasDF.tail(1))
(18) Spark Jobs
robberyLosAngelesDF: pyspark.sql.dataframe.DataFrame = [id: string, dateReported: timestamp ... 17 more fields]
df1: pyspark.sql.dataframe.DataFrame = [id: string, dateReported: timestamp ... 17 more fields]
robberyLosAngelesDF1: pyspark.sql.dataframe.DataFrame = [timeOccurred: integer, crimeCodeDescription: string]
robberiesByMonthLosAngelesDF: pyspark.sql.dataframe.DataFrame = [timeOccurred: integer, count: long]
robberiesPhiladelphiaDF: pyspark.sql.dataframe.DataFrame = [district: integer, dispatch_date_time: timestamp ... 12 more fields]
robberiesPhiladelphiaDF1: pyspark.sql.dataframe.DataFrame = [district: integer, dispatch_date_time: integer ... 12 more fields]
robberiesPhiladelphiaDF2: pyspark.sql.dataframe.DataFrame = [dispatch_date_time: integer, ucr_general_description: string]
robberiesByMonthPhiladelphiaDF: pyspark.sql.dataframe.DataFrame = [dispatch_date_time: integer, count: long]
robberyDallasDF: pyspark.sql.dataframe.DataFrame = [incidentNumberWithYear: string, incidentNumberWithoutYear: integer ... 80 more fields]
robberiesByMonthDallasDF: pyspark.sql.dataframe.DataFrame = [monthOccurred: string, count: long]
(Los Angeles) Highest is [Row(timeOccurred=12, count=853)]
(Los Angeles) Lowest is [Row(timeOccurred=2, count=675)]
(Philadelphia) Highest is [Row(dispatch_date_time=10, count=572)]
(Philadelphia) Lowest is [Row(dispatch_date_time=2, count=416)]
(Dallas) Highest is [Row(monthOccurred='01', count=743)]
(Dallas) Lowest is [Row(monthOccurred='03', count=410)]
Command took 55.40 seconds -- by kurray1@mail.uc.edu at 13/4/2023, 9:47:33 pm on Yasaswi_CC_Cluster
Cmd 18
Search
ENG IN
22:51 16-04-2023

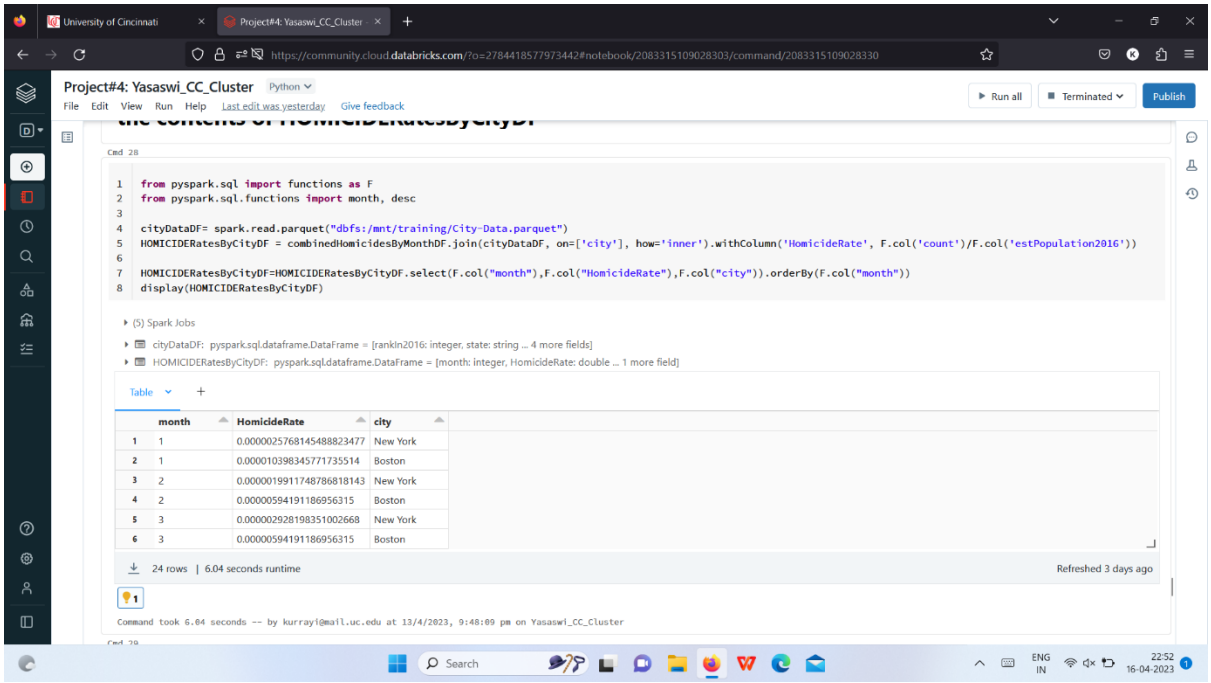
University of Cincinnati
Project#4: Yasaswi_CC_Cluster
https://community.cloud.databricks.com/?o=2784418577973442#notebook/2083315109028303/command/2083315109028330
Project#4: Yasaswi_CC_Cluster
Python
File Edit View Run Help Last edit was yesterday Give feedback
Run all
Terminated
Publish
Hint: You may want to apply the union() method in this example to combine the three datasets.
Cmd 19
Hint: Combined 3 Cities
Cmd 20
# TODO
from pyspark.sql.functions import lit
robberiesByMonthLosAngelesDF = robberiesByMonthLosAngelesDF.withColumn("city",lit("Los Angeles"))
robberiesByMonthPhiladelphiaDF = robberiesByMonthPhiladelphiaDF.withColumn("city",lit("Philadelphia"))
robberiesByMonthDallasDF = robberiesByMonthDallasDF.withColumn("city",lit("Dallas"))
combinedRobberiesByMonthDF = robberiesByMonthDallasDF.union(robberiesByMonthLosAngelesDF).union(robberiesByMonthPhiladelphiaDF)
high = combinedRobberiesByMonthDF.sort(desc('count')).head(1)
low = combinedRobberiesByMonthDF.sort(desc('count')).tail(1)
print("Highest is at : ", high)
print("Lowest is at : ", low)
(10) Spark Jobs
robberiesByMonthLosAngelesDF: pyspark.sql.dataframe.DataFrame = [timeOccurred: integer, count: long ... 1 more field]
robberiesByMonthPhiladelphiaDF: pyspark.sql.dataframe.DataFrame = [dispatch_date_time: integer, count: long ... 1 more field]
robberiesByMonthDallasDF: pyspark.sql.dataframe.DataFrame = [monthOccurred: string, count: long ... 1 more field]
combinedRobberiesByMonthDF: pyspark.sql.dataframe.DataFrame = [monthOccurred: string, count: long ... 1 more field]
Highest is at : [Row(monthOccurred='12', count=853, city='Los Angeles')]
Lowest is at : [Row(monthOccurred='03', count=410, city='Dallas')]
Command took 48.35 seconds -- by kurray1@mail.uc.edu at 13/4/2023, 9:47:41 pm on Yasaswi_CC_Cluster
Cmd 21
Search
ENG IN
22:51 16-04-2023

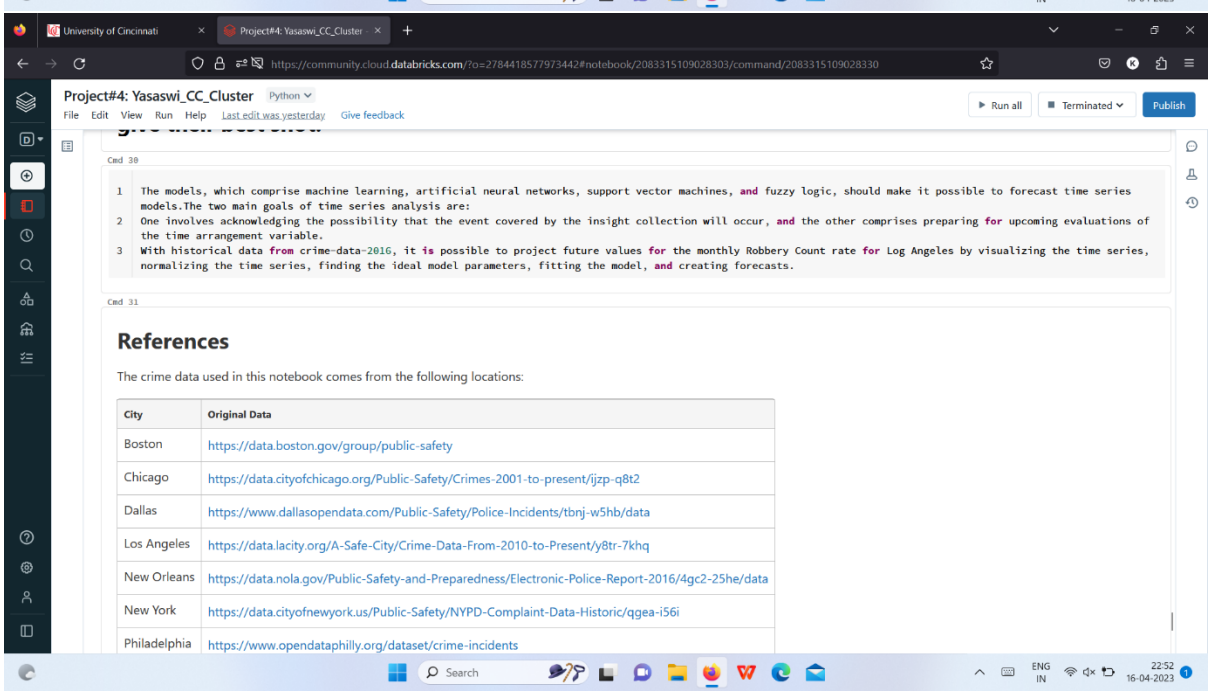

University of Cincinnati
Project#4: Yasaswi_CC_Cluster
https://community.cloud.databricks.com/?o=2784418577973442#notebook/2083315109028303/command/2083315109028330
Project#4: Yasaswi_CC_Cluster
Python
File Edit View Run Help Last edit was yesterday Give feedback
Run all
Terminated
Publish
9
10 Philadephiadata =
crimeDataPhiladelphiaDF.withColumn("monthOccurred",split(col("dispatch_date"),"-").getItem(1)).withColumn("crime",lower(col("ucr_general_description")))
11 robberiesByMonthPhiladelphiaDF = Philadephiadata.select("monthOccurred","crime")
12 robberiesByMonthPhiladelphiaDF = robberiesByMonthPhiladelphiaDF.filter("crime like '%robbery%'")
13
14 Dallasdata = crimeDataDallasDF.withColumn("monthOccurred",split(col("callReceivedDateTime"),"/").getItem(0)).withColumn("crime",lower(col("typeOfIncident")))
15 robberiesByMonthDallasDF = Dallasdata.select("monthOccurred","crime")
16 robberiesByMonthDallasDF = robberiesByMonthDallasDF.filter("crime like '%robbery%'")
17
18 robbery_per_month_dallas_city = robberiesByMonthDallasDF.groupBy("monthOccurred").count().withColumn("city",lit("Dallas"))
19 robbery_per_month_losangels_city = robberiesByMonthLosAngelesDF.groupBy("monthOccurred").count().withColumn("city",lit("LosAngeles"))
20 robbery_per_month_philadelphia_city = robberiesByMonthPhiladelphiaDF.groupBy("monthOccurred").count().withColumn("city",lit("Philadelphia"))
21 combinedRobberiesByMonthDF =
robbery_per_month_dallas_city.union(robbery_per_month_losangels_city).union(robbery_per_month_philadelphia_city).withColumnRenamed("count","robbery").sort("monthOccurred",ascending=True)
22 display(combinedRobberiesByMonthDF)
(4) Spark Jobs
LosAngelsData: pyspark.sql.dataframe.DataFrame = [id: string, dateReported: timestamp ... 19 more fields]
robberiesByMonthLosAngelesDF: pyspark.sql.dataframe.DataFrame = [monthOccurred: string, crime: string]
Philadephiadata: pyspark.sql.dataframe.DataFrame = [district: integer, dispatch_date_time: timestamp ... 14 more fields]
robberiesByMonthPhiladelphiaDF: pyspark.sql.dataframe.DataFrame = [monthOccurred: string, crime: string]
Dallasdata: pyspark.sql.dataframe.DataFrame = [incidentNumberWithYear: string, incidentNumberWithoutYear: integer ... 80 more fields]
robberiesByMonthDallasDF: pyspark.sql.dataframe.DataFrame = [monthOccurred: string, crime: string]
robbery_per_month_dallas_city: pyspark.sql.dataframe.DataFrame = [monthOccurred: string, count: long ... 1 more field]
robbery_per_month_losangels_city: pyspark.sql.dataframe.DataFrame = [monthOccurred: string, count: long ... 1 more field]
robbery_per_month_philadelphia_city: pyspark.sql.dataframe.DataFrame = [monthOccurred: string, count: long ... 1 more field]
combinedRobberiesByMonthDF: pyspark.sql.dataframe.DataFrame = [monthOccurred: string, robbery: long ... 1 more field]
Table
Search
ENG IN
22:52 16-04-2023

University of Cincinnati
Project#4: Yasaswi_CC_Cluster
https://community.cloud.databricks.com/?o=2784418577973442#notebook/2083315109028303/command/2083315109028330
Project#4: Yasaswi_CC_Cluster
Python
File Edit View Run Help Last edit was yesterday Give feedback
Run all
Terminated
Publish
combinedRobberiesByMonthDF: pyspark.sql.dataframe.DataFrame = [monthOccurred: string, robbery: long ... 1 more field]
Table
monthOccurred robbery city
1 01 743 Dallas
2 01 835 LosAngeles
3 01 524 Philadelphia
4 02 757 LosAngeles
5 02 439 Dallas
6 02 412 Philadelphia
36 rows | 24.35 seconds runtime
Refreshed 3 days ago
Command took 24.35 seconds -- by kurray1@mail.uc.edu at 13/4/2023, 9:47:51 pm on Yasaswi_CC_Cluster
Cmd 23
Question 7: Find the "per capita robbery rates" using ther Hint below, and plot graph as above for the per capita robbery rates per month for each of the three cities, producing one Graph using the contents of robberyRatesByCityDF :
Los Angeles
Philadelphia
Dallas
While the above graph is interesting, it's flawed: it's comparing the raw numbers of robberies, not the per capita robbery rates.
Search
ENG IN
22:52 16-04-2023


University of Cincinnati
Project#4: Yasaswi_CC_Cluster
https://community.cloud.databricks.com/?o=2784418577973442#notebook/2083315109028303/command/2083315109028330
Project#4: Yasaswi_CC_Cluster
Python
File Edit View Run Help Last edit was yesterday Give feedback
Run all
Terminated
Publish
8 population_LosAngeles = cityDataDF.filter("city = 'Los Angeles'").select("estPopulation2016").collect()[0]["estPopulation2016"]
9 population_Philadelphia = cityDataDF.filter("city = 'Philadelphia'").select("estPopulation2016").collect()[0]["estPopulation2016"]
10 population_Dallas = cityDataDF.filter("city = 'Dallas'").select("estPopulation2016").collect()[0]["estPopulation2016"]
11 robbery_rates_for_LA = combinedRobberiesByMonthDF.withColumn("robberyRate",(combinedRobberiesByMonthDF.robbery)/population_LosAngeles).where("city = 'LosAngeles'")
12 robbery_rates_for_philadelphia = combinedRobberiesByMonthDF.withColumn("robberyRate",(combinedRobberiesByMonthDF.robbery)/population_Philadelphia).where("city = 'Philadelphia'")
13 robbery_rates_for_Dallas = combinedRobberiesByMonthDF.withColumn("robberyRate",(combinedRobberiesByMonthDF.robbery)/population_Dallas).where("city = 'Dallas'")
14 robberyRatesByCityDF = robbery_rates_for_LA.union(robbery_rates_for_philadelphia).union(robbery_rates_for_Dallas).sort("monthOccurred",ascending=True)
15 display(robberyRatesByCityDF)
(8) Spark Jobs
cityDataDF: pyspark.sql.dataframe.DataFrame = [rankIn2016: integer, state: string ... 4 more fields]
robbery_rates_for_LA: pyspark.sql.dataframe.DataFrame = [monthOccurred: string, robbery: long ... 2 more fields]
robbery_rates_for_philadelphia: pyspark.sql.dataframe.DataFrame = [monthOccurred: string, robbery: long ... 2 more fields]
robbery_rates_for_Dallas: pyspark.sql.dataframe.DataFrame = [monthOccurred: string, robbery: long ... 2 more fields]
robberyRatesByCityDF: pyspark.sql.dataframe.DataFrame = [monthOccurred: string, robbery: long ... 2 more fields]
Table
monthOccurred robbery city robberyRate
1 01 743 Dallas 0.0005637632983263893
2 01 835 LosAngeles 0.0002099930538824572
3 01 524 Philadelphia 0.0003342109559964079
4 02 757 LosAngeles 0.0001903769362742755 8
5 02 439 Dallas 0.0003330983687285 1266
6 02 412 Philadelphia 0.0002627765531880 1535
36 rows | 25.61 seconds runtime
Refreshed 3 days ago
Command took 25.61 seconds -- by kurrayi@mail.uc.edu at 13/4/2023, 9:47:57 pm on Yasaswi_CC_Cluster
Search
ENG IN
22:52 16-04-2023


University of Cincinnati
Project#4: Yasaswi_CC_Cluster
https://community.cloud.databricks.com/?o=2784418577973442#notebook/2083315109028303/command/2083315109028330
Project#4: Yasaswi_CC_Cluster
Python
File Edit View Run Help Last edit was yesterday Give feedback
Run all
Terminated
Publish
26
27 combinedHomicidesByMonthDF = reduce(DataFrame.unionAll, dfs)
28 combinedHomicidesByMonthDF1=combinedHomicidesByMonthDF.sort(desc('count'))
29 print(combinedHomicidesByMonthDF1.head(1))
30 print(combinedHomicidesByMonthDF1.tail(1))
(10) Spark Jobs
crimeDataNewyorkDF: pyspark.sql.dataframe.DataFrame = [complaintNumber: integer, keyCode: integer ... 16 more fields]
homicideNewyorkDF: pyspark.sql.dataframe.DataFrame = [complaintNumber: integer, keyCode: integer ... 16 more fields]
homicideNewyorkDF1: pyspark.sql.dataframe.DataFrame = [complaintNumber: integer, keyCode: integer ... 17 more fields]
homicideByMonthNewyorkDF: pyspark.sql.dataframe.DataFrame = [month: integer, offenseDescription: string]
homicideByMonthNewyorkDF1: pyspark.sql.dataframe.DataFrame = [month: integer, count: long]
crimeDataBostonDF: pyspark.sql.dataframe.DataFrame = [INCIDENT_NUMBER: string, OFFENSE_CODE: integer ... 15 more fields]
homicideBostonDF: pyspark.sql.dataframe.DataFrame = [INCIDENT_NUMBER: string, OFFENSE_CODE: integer ... 15 more fields]
homicideByMonthBostonDF: pyspark.sql.dataframe.DataFrame = [MONTH: integer, OFFENSE_CODE_GROUP: string]
homicideByMonthBostonDF1: pyspark.sql.dataframe.DataFrame = [MONTH: integer, count: long]
homicideByMonthNewyorkDF2: pyspark.sql.dataframe.DataFrame = [month: integer, count: long ... 1 more field]
homicideByMonthBostonDF2: pyspark.sql.dataframe.DataFrame = [MONTH: integer, count: long ... 1 more field]
combinedHomicidesByMonthDF: pyspark.sql.dataframe.DataFrame = [month: integer, count: long ... 1 more field]
combinedHomicidesByMonthDF1: pyspark.sql.dataframe.DataFrame = [month: integer, count: long ... 1 more field]
[Row(month=8, count=36, city='New York')]
[Row(month=3, count=4, city='Boston')]
Command took 10.13 seconds -- by kurrayi@mail.uc.edu at 13/4/2023, 9:48:04 pm on Yasaswi_CC_Cluster
Cmd 27
Question 9: Find the "per capita HOMICIDE rates" using the Hint in Qn 7, and plot graph as above for the per capita HOMICIDE rates per month for each of the two cities, producing one Graph using
Search
ENG IN
22:52 16-04-2023


University of Cincinnati
Project#4: Yasaswi_CC_Cluster
https://community.cloud.databricks.com/?o=2784418577973442#notebook/2083315109028303/command/2083315109028330
Project#4: Yasaswi_CC_Cluster
Python
File Edit View Run Help Last edit was yesterday Give feedback
Run all
Terminated
Publish
Cmd 28
1 from pyspark.sql import functions as F
2 from pyspark.sql.functions import month, desc
3
4 cityDataDF= spark.read.parquet("dbfs:/mnt/training/City-Data.parquet")
5 HOMICIDERatesByCityDF = combinedHomicidesByMonthDF.join(cityDataDF, on=['city'], how='inner').withColumn('HomicideRate', F.col('count')/F.col('estPopulation2016'))
6
7 HOMICIDERatesByCityDF=HOMICIDERatesByCityDF.select(F.col("month"),F.col("HomicideRate"),F.col("city")).orderBy(F.col("month"))
8 display(HOMICIDERatesByCityDF)
(5) Spark Jobs
cityDataDF: pyspark.sql.dataframe.DataFrame = [rankIn2016: integer, state: string ... 4 more fields]
HOMICIDERatesByCityDF: pyspark.sql.dataframe.DataFrame = [month: integer, HomicideRate: double ... 1 more field]
Table
month HomicideRate city
1 1 0.00000257681454888234 77 New York
2 1 0.00001039834577173551 4 Boston
3 2 0.00000199117487868181 43 New York
4 2 0.00000594191186956315 Boston
5 3 0.00000292819835100266 8 New York
6 3 0.00000594191186956315 Boston
24 rows | 6.04 seconds runtime
Refreshed 3 days ago
Command took 6.04 seconds -- by kurrayi@mail.uc.edu at 13/4/2023, 9:48:09 pm on Yasaswi_CC_Cluster
Cmd 29
Search
ENG IN
22:52 16-04-2023


University of Cincinnati
Project#4: Yasaswi_CC_Cluster
https://community.cloud.databricks.com/?o=2784418577973442#notebook/2083315109028303/command/2083315109028330
Project#4: Yasaswi_CC_Cluster
Python
File Edit View Run Help Last edit was yesterday Give feedback
Run all
Terminated
Publish
Cmd 30
1 The models, which comprise machine learning, artificial neural networks, support vector machines, and fuzzy logic, should make it possible to forecast time series
models.The two main goals of time series analysis are:
2 One involves acknowledging the possibility that the event covered by the insight collection will occur, and the other comprises preparing for upcoming evaluations of
the time arrangement variable.
3 With historical data from crime-data-2016, it is possible to project future values for the monthly Robbery Count rate for Log Angeles by visualizing the time series,
normalizing the time series, finding the ideal model parameters, fitting the model, and creating forecasts.
Cmd 31
References
The crime data used in this notebook comes from the following locations:
City
Original Data
Boston
https://data.boston.gov/group/public-safety
Chicago
https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2
Dallas
https://www.dallasopendata.com/Public-Safety/Police-Incidents/tbnj-w5hb/data
Los Angeles
https://data.lacity.org/A-Safe-City/Crime-Data-From-2010-to-Present/y8tr-7khq
New Orleans
https://data.nola.gov/Public-Safety-and-Preparedness/Electronic-Police-Report-2016/4gc2-25he/data
New York
https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Historic/qgea-i56i
Philadelphia
https://www.opendataphilly.org/dataset/crime-incidents
Search
ENG IN
22:52 16-04-2023