VIVEKANAND EDUCATION SOCIETY'S
INSTITUTE OF TECHNOLOGY

Department of Computer Engineering

Computational Lab - I
Mini Project Report on

AADHAR DATA ANALYSIS

Submitted in partial fulfilment of the requirements
of FOURTH Year Computer Engineering

By

Siddhesh Gadge D17B-13
Nachiket Joag D17B-27
Anand Pal D17B-41

Supervisor: Mrs. Richa Sharma

DEPARTMENT OF COMPUTER ENGINEERING
V.E.S INSTITUTE OF TECHNOLOGY

2021-22

AADHAR DATA ANALYSIS

1. Problem Definition and Scope of Project

1.1 Introduction

Debates around Aadhaar have tended to be polarised—yet national household data has been thin on what Aadhaar has done (and not done) for the ordinary residents of India.

In what ways has Aadhaar empowered or excluded them? To what extent do they trust and use the identification system? In which aspects is it serving them well or poorly—or not at all? Our study set out to answer some of these questions with data.

TRENDS

How many are enrolled?
Who is not yet enrolled—and why?
How many updates are needed, i.e., how common are errors?
Do errors get corrected? What is the update experience?
And how easy or difficult is the process?
How widely and how frequently is Aadhaar used?
What is the experience of using Aadhaar for key services (PDS,
MGNREGS, social pensions, SIM cards, and bank accounts)?
If residents face problems with Aadhaar, how does that affect their access to services?
What benefits and challenges do people see?
How satisfied are people with Aadhaar overall?
Do they trust the system?

1.2 Problem definition and scope of project

The primary purpose of our project was to give a broad cross-section of Indian residents a voice in the national discourse on Aadhaar.This study distils insights drawn from two national household surveys on Aadhaar, conducted between May and September 2019, and subsequent human-centred design research. Capturing the experiences and perspectives of over 167,000 residents, together these surveys represent the largest primary dataset on the use of Aadhaar and, more broadly, digital ID anywhere in the world. We believe the success of Aadhaar will ultimately depend on how well the program can learn from the experiences and concerns of those who use (or are unable to use) Aadhaar across a wide range of circumstances in their daily lives. Taking residents' perspectives into account can help better design and implement Aadhaar.Our aim was also to help identify which aspects of Aadhaar are working and are not working, to what extent and for whom.We hope these findings will inform the efforts of policymakers to further the objective of a universal identity across India and allow Aadhaar to live up to its stated aspirations.

1.3 Dataset used

The data comes from the stateofaadhar.in website, which conducted a poll in 2019 by asking questions such as where people utilised adhar for the most fundamental information, and so on.

The dataset is made up of 35 columns and 575127 rows, and it was compiled from a survey conducted by the state of Aadhar in India in 2019.

```
RangeIndex: 575127 entries, 0 to 575126
Data columns (total 35 columns):
 #   Column                                      Non-Null Count   Dtype
---  ------                                      --------------   -----
 0   hh_id                                       575127 non-null  int64
 1   country                                     575127 non-null  object
 2   state                                       575127 non-null  object
 3   district_name                               575127 non-null  object
 4   region_type                                 575127 non-null  object
 5   state_of_origin                             575127 non-null  object
 6   town_village                                575127 non-null  object
 7   gender                                      575127 non-null  object
 8   literacy                                    575127 non-null  object
 9   occupation                                  575127 non-null  object
 10  education                                   575127 non-null  object
 11  employment_status                           575127 non-null  object
 12  has_aadhaar                                 575127 non-null  object
 13  correct_name                                575127 non-null  object
 14  correct_dob                                 575127 non-null  object
 15  correct_gender                              575127 non-null  object
 16  correct_address                             575127 non-null  object
 17  correct_photo                               575127 non-null  object
 18  correct_mobile                              575127 non-null  object
 19  used_other_employment_schemes               575127 non-null  object
 20  used_scholarships                           575127 non-null  object
 21  used_open_bank_account                      575127 non-null  object
 22  used_insurance                              575127 non-null  object
 23  used_debt_loan                              575127 non-null  object
 24  used_sim_phone                              575127 non-null  object
 25  used_school_college                         575127 non-null  object
 26  used_get_another_id                         575127 non-null  object
 27  used_job_application                        575127 non-null  object
 28  used_age_proof                              575127 non-null  object
 29  used_land_vehicle_house_marriage_registratio 575127 non-null  object
 30  cal_age_in_yrs                               575127 non-null  float64
 31  cal_age_type                                 575127 non-null  object
 32  cal_employment                               575127 non-null  object
 33  cal_education_yrs                            575127 non-null  int64
 34  cal_correct_card                             575127 non-null  object
```

2. LITERATURE REVIEW

[1] provides a systematic review of the materials/articles available through secondary sources such as newspapers, research papers and government reports on the Aadhaar project. An attempt has been made through this study to understand the planning and implementation stage of the Aadhaar Project till 2014. The study also attempts to identify potential risks.and suggest a contingency plan for this and similar government projects in future, to ensure a better success rate. This study identifies various gaps and recommends a plan of action as well as appropriate process changes to enhance project success of Aadhaar Project in future. Based on the insights from this study a model to enhance the success rate of similar projects has also been proposed.

[2] throws light on the problems faced by the people for linking their aadhaar card with public services and the possible potential solutions.

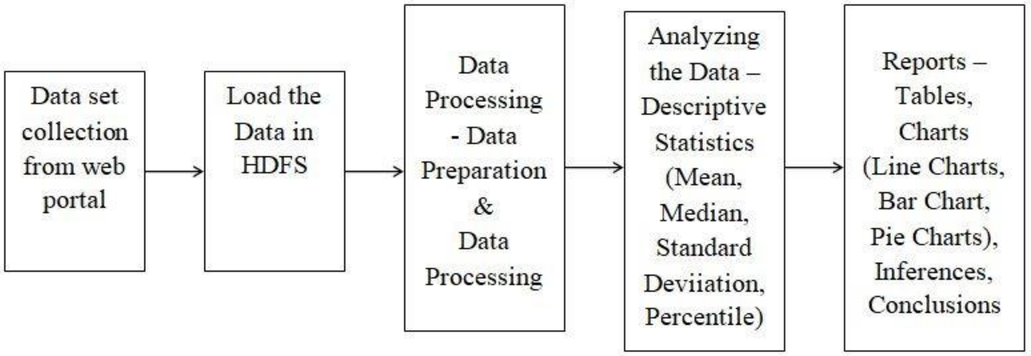
[3] analyzes the Aadhaar card data set against different research queries for example total number of aadhaar cards approved by state, rejected by state, total number of aadhaar card applicants by gender and total number of aadhaar card applicants by age type.

[4] focuses on the journey of Aadhaar from its history to the current condition. The paper also describes the authentication process, and the updates happened over time. It also provides an analysis of the security attacks witnessed so far as well as the possible countermeasure and its classification.

[5] presents a brief review on Aadhaar card, and discusses the scope and advantages of linking Aadhaar card to various systems. Further, it also presents various cases in which Aadhaar cards may pose security threats. The observations of the Supreme Court of India are also presented in this paper followed by a discussion on the loopholes in the existing system.

3. CONCEPTUAL SYSTEM DESIGN

3.1 CSD diagram


Data set collection from web portal
Load the Data in HDFS
Data Processing - Data Preparation & Data Processing
Analyzing the Data – Descriptive Statistics (Mean, Median, Standard Deviiation, Percentile)
Reports – Tables, Charts (Line Charts, Bar Chart, Pie Charts), Inferences, Conclusions

3.2 Methodology

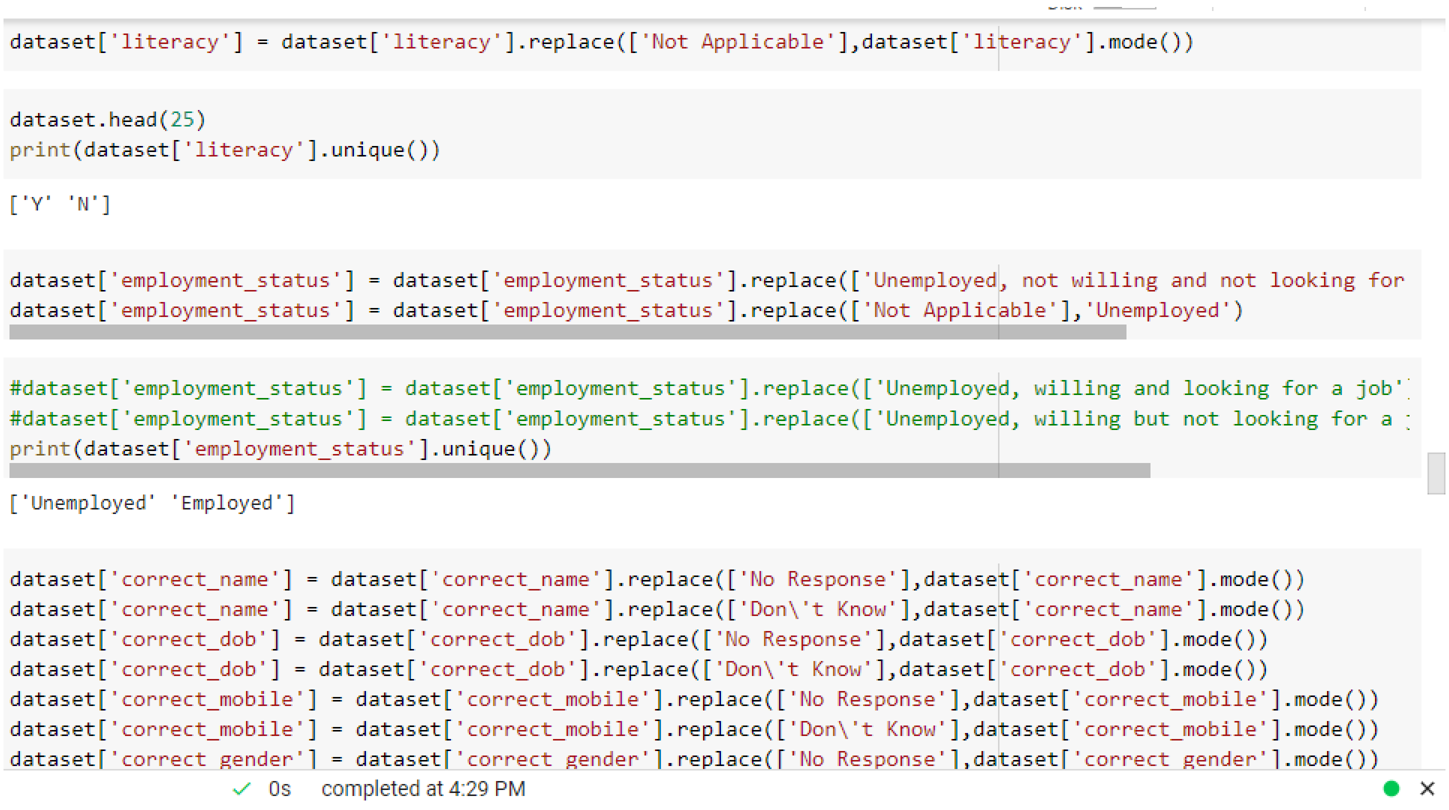
3.2.1 Data gathering / Loading

The data comes from the stateofaadhar.in website, which conducted a poll in 2019 by asking questions such as where people utilised adhar for the most fundamental information, and so on.

3.2.2 Data preprocessing and descriptive analysis

All the data has been explored._The dataset is made up of 35 columns and 575127 rows, and it was compiled from a survey conducted by the state of Aadhar in India in 2019.

All not applicable no response field were replaced by mode of that particular column



```
dataset['literacy'] = dataset['literacy'].replace(['Not Applicable'],dataset['literacy'].mode())
```

```
dataset.head(25)
print(dataset['literacy'].unique())
```

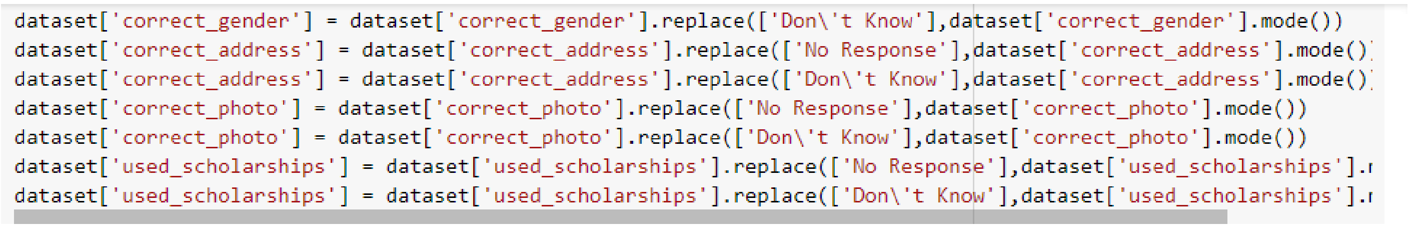
```
['Y' 'N']
```

```
dataset['employment_status'] = dataset['employment_status'].replace(['Unemployed, not willing and not looking for
dataset['employment_status'] = dataset['employment_status'].replace(['Not Applicable'],'Unemployed')
```

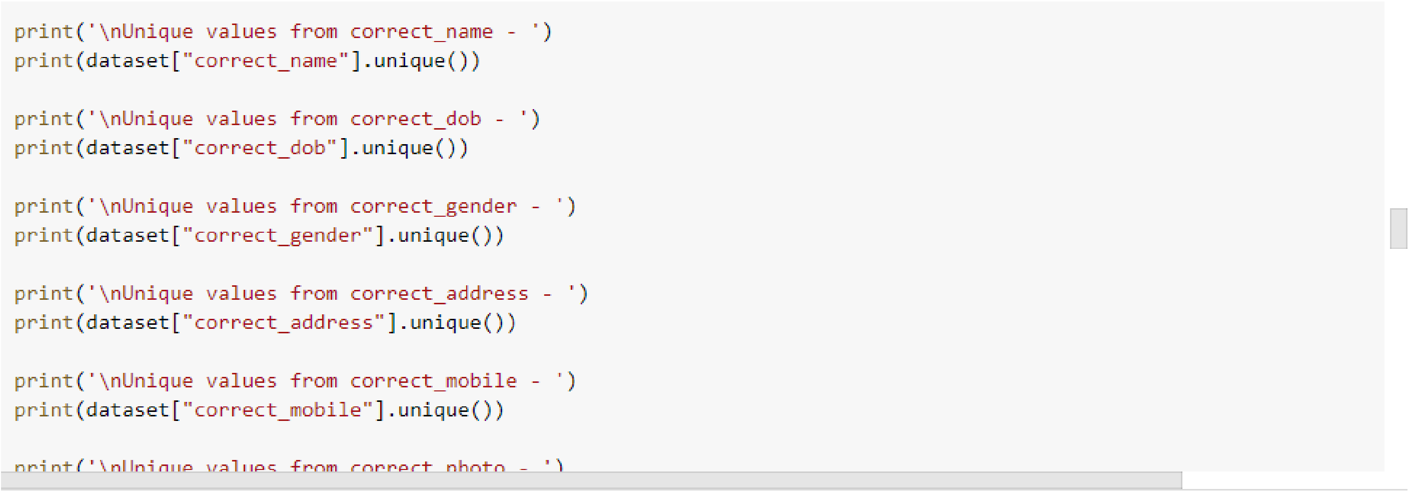
```
#dataset['employment_status'] = dataset['employment_status'].replace(['Unemployed, willing and looking for a job'
#dataset['employment_status'] = dataset['employment_status'].replace(['Unemployed, willing but not looking for a
print(dataset['employment_status'].unique())
```

```
['Unemployed' 'Employed']
```

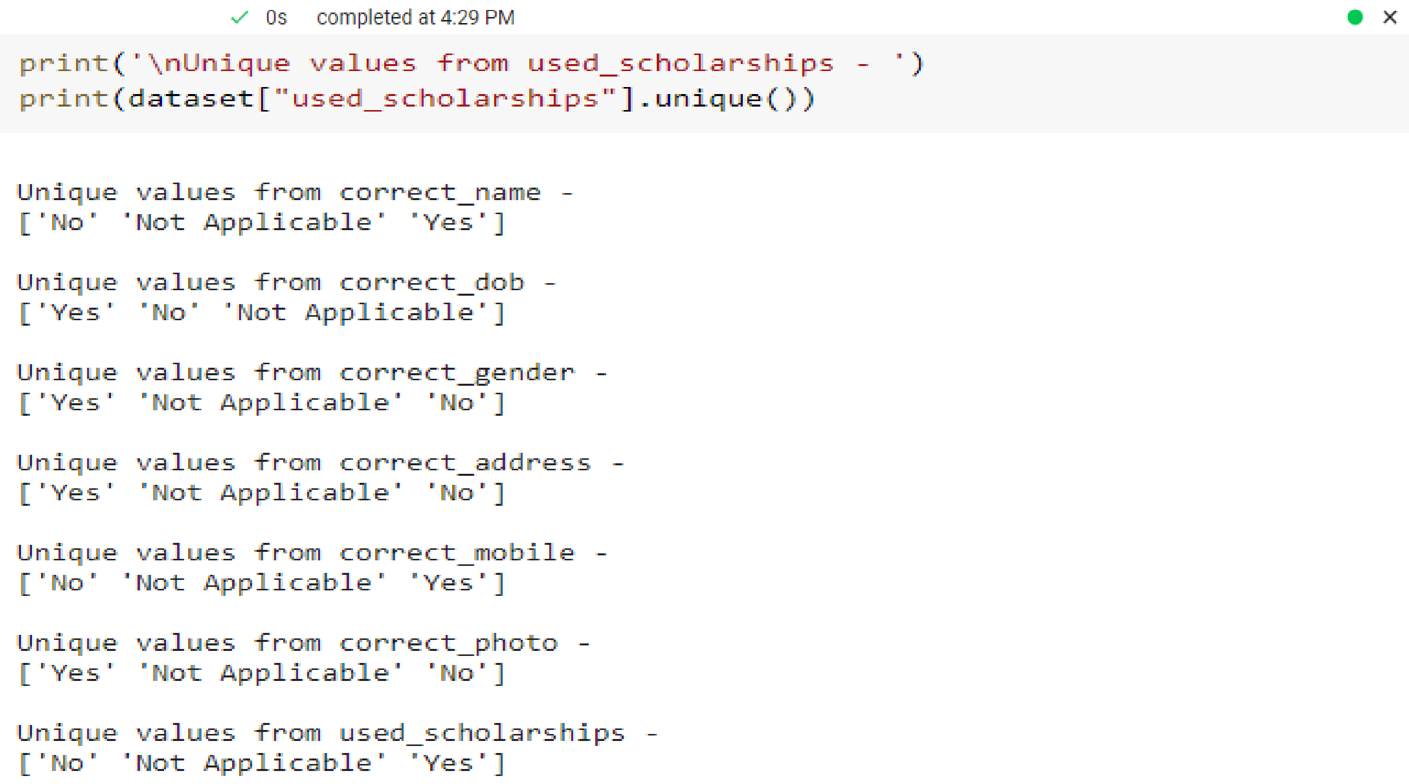
```
dataset['correct_name'] = dataset['correct_name'].replace(['No Response'],dataset['correct_name'].mode())
dataset['correct_name'] = dataset['correct_name'].replace(['Don\'t Know'],dataset['correct_name'].mode())
dataset['correct_dob'] = dataset['correct_dob'].replace(['No Response'],dataset['correct_dob'].mode())
dataset['correct_dob'] = dataset['correct_dob'].replace(['Don\'t Know'],dataset['correct_dob'].mode())
dataset['correct_mobile'] = dataset['correct_mobile'].replace(['No Response'],dataset['correct_mobile'].mode())
dataset['correct_mobile'] = dataset['correct_mobile'].replace(['Don\'t Know'],dataset['correct_mobile'].mode())
dataset['correct_gender'] = dataset['correct_gender'].replace(['No Response'],dataset['correct_gender'].mode())
```



```
dataset['correct_gender'] = dataset['correct_gender'].replace(['Don\'t Know'],dataset['correct_gender'].mode())
dataset['correct_address'] = dataset['correct_address'].replace(['No Response'],dataset['correct_address'].mode()
dataset['correct_address'] = dataset['correct_address'].replace(['Don\'t Know'],dataset['correct_address'].mode()
dataset['correct_photo'] = dataset['correct_photo'].replace(['No Response'],dataset['correct_photo'].mode())
dataset['correct_photo'] = dataset['correct_photo'].replace(['Don\'t Know'],dataset['correct_photo'].mode())
dataset['used_scholarships'] = dataset['used_scholarships'].replace(['No Response'],dataset['used_scholarships'].r
dataset['used_scholarships'] = dataset['used_scholarships'].replace(['Don\'t Know'],dataset['used_scholarships'].r
```

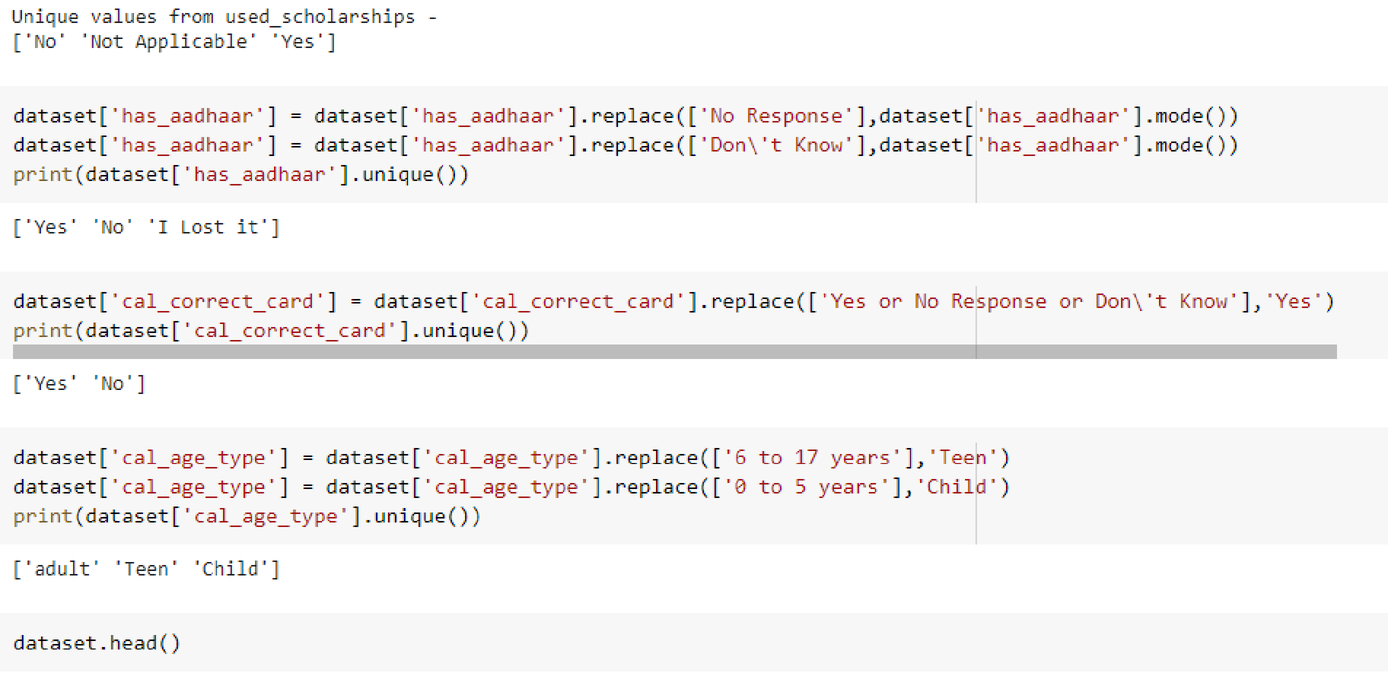


```
print('\nUnique values from correct_name - ')
print(dataset["correct_name"].unique())

print('\nUnique values from correct_dob - ')
print(dataset["correct_dob"].unique())

print('\nUnique values from correct_gender - ')
print(dataset["correct_gender"].unique())

print('\nUnique values from correct_address - ')
print(dataset["correct_address"].unique())

print('\nUnique values from correct_mobile - ')
print(dataset["correct_mobile"].unique())

print('\nUnique values from correct_photo - ')
```



```
print('\nUnique values from used_scholarships - ')
print(dataset["used_scholarships"].unique())
```

```
Unique values from correct_name -
['No' 'Not Applicable' 'Yes']

Unique values from correct_dob -
['Yes' 'No' 'Not Applicable']

Unique values from correct_gender -
['Yes' 'Not Applicable' 'No']

Unique values from correct_address -
['Yes' 'Not Applicable' 'No']

Unique values from correct_mobile -
['No' 'Not Applicable' 'Yes']

Unique values from correct_photo -
['Yes' 'Not Applicable' 'No']

Unique values from used_scholarships -
['No' 'Not Applicable' 'Yes']
```



```
Unique values from used_scholarships -
['No' 'Not Applicable' 'Yes']
```

```
dataset['has_aadhaar'] = dataset['has_aadhaar'].replace(['No Response'],dataset['has_aadhaar'].mode())
dataset['has_aadhaar'] = dataset['has_aadhaar'].replace(['Don\'t Know'],dataset['has_aadhaar'].mode())
print(dataset['has_aadhaar'].unique())
```

```
['Yes' 'No' 'I Lost it']
```

```
dataset['cal_correct_card'] = dataset['cal_correct_card'].replace(['Yes or No Response or Don\'t Know'],'Yes')
print(dataset['cal_correct_card'].unique())
```

```
['Yes' 'No']
```

```
dataset['cal_age_type'] = dataset['cal_age_type'].replace(['6 to 17 years'],'Teen')
dataset['cal_age_type'] = dataset['cal_age_type'].replace(['0 to 5 years'],'Child')
print(dataset['cal_age_type'].unique())
```

```
['adult' 'Teen' 'Child']
```

```
dataset.head()
```

3.2.3 Filtering

No filtering was performed on the dataset as all the data was used and analyzed both separately and collectively.

3.2.4 Classification / Clustering

This analysis does not involve classification and clustering data.

3.2.5 Visualizations

The results of analysis were visualized using various Python libraries like matplotlib, seaborn, etc. were used to obtain charts and graphs.

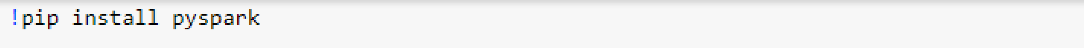
4. TECHNOLOGY USED

In this analysis, a combination of PySpark and Python on Google Colab have been used for exploring and analysing the dataset, and visualizing data and results.
Also Hadoop and Hive has been used for analysis and visualization.
Additional libraries have been used to achieve required results, some of which are mentioned below.

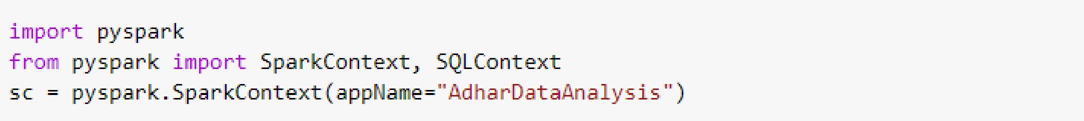
1. SparkContext
2. SQLContext
3. Pandas
4. Numpy
5. Seaborn
6. Matplotlib

5. IMPLEMENTATION

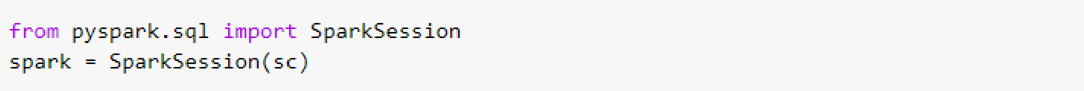
Library Imports –



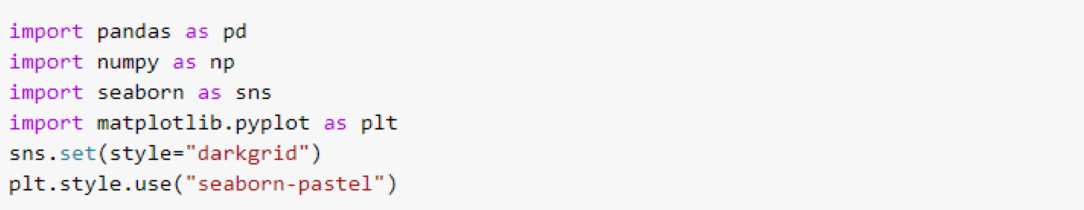
```
!pip install pyspark
```



```
import pyspark
from pyspark import SparkContext, SQLContext
sc = pyspark.SparkContext(appName="AdharDataAnalysis")
```

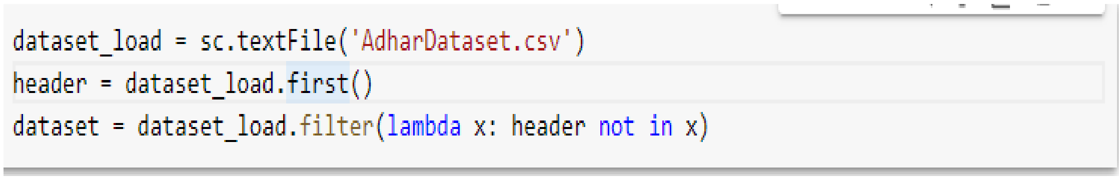


```
from pyspark.sql import SparkSession
spark = SparkSession(sc)
```



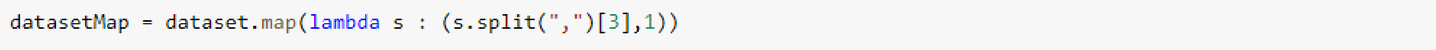
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
sns.set(style="darkgrid")
plt.style.use("seaborn-pastel")
```

The Dataset is loaded and Header is removed-

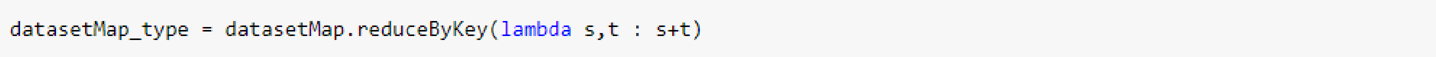


```
dataset_load = sc.textFile('AdharDataset.csv')
header = dataset_load.first()
dataset = dataset_load.filter(lambda x: header not in x)
```

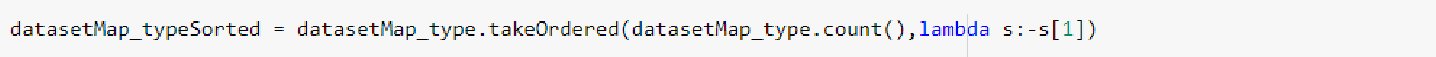
List Name of the countries with their count –



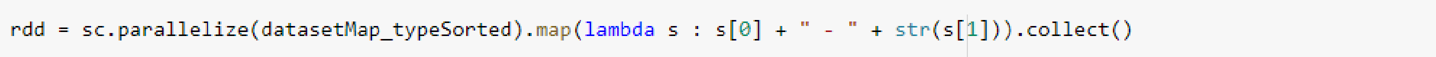
```
datasetMap = dataset.map(lambda s : (s.split(",")[3],1))
```



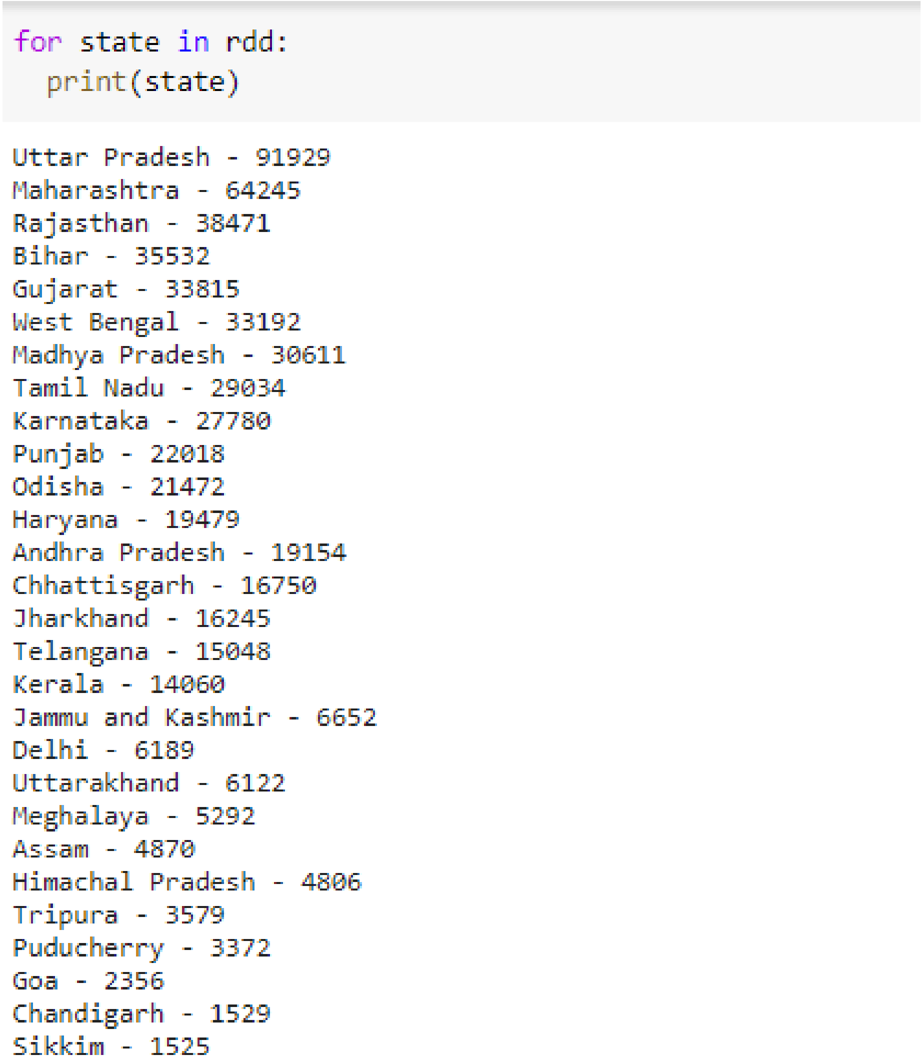
```
datasetMap_type = datasetMap.reduceByKey(lambda s,t : s+t)
```



```
datasetMap_typeSorted = datasetMap_type.takeOrdered(datasetMap_type.count(),lambda s:-s[1])
```



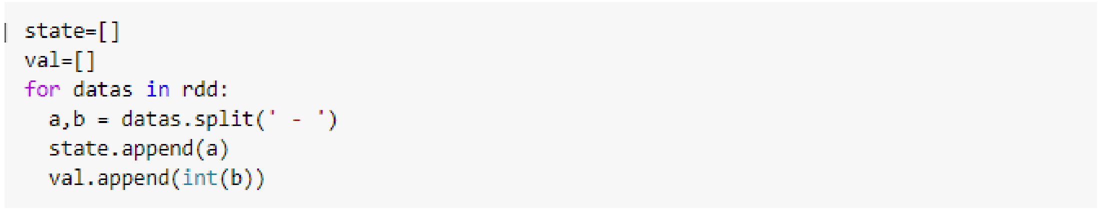
```
rdd = sc.parallelize(datasetMap_typeSorted).map(lambda s : s[0] + " - " + str(s[1])).collect()
```



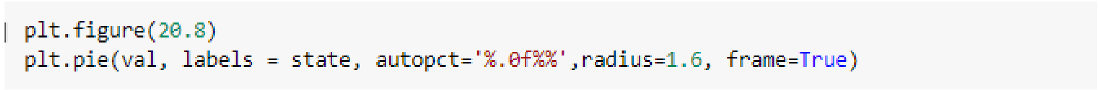
```
for state in rdd:
  print(state)
```

```
Uttar Pradesh - 91929
Maharashtra - 64245
Rajasthan - 38471
Bihar - 35532
Gujarat - 33815
West Bengal - 33192
Madhya Pradesh - 30611
Tamil Nadu - 29034
Karnataka - 27780
Punjab - 22018
Odisha - 21472
Haryana - 19479
Andhra Pradesh - 19154
Chhattisgarh - 16750
Jharkhand - 16245
Telangana - 15048
Kerala - 14060
Jammu and Kashmir - 6652
Delhi - 6189
Uttarakhand - 6122
Meghalaya - 5292
Assam - 4870
Himachal Pradesh - 4806
Tripura - 3579
Puducherry - 3372
Goa - 2356
Chandigarh - 1529
Sikkim - 1525
```

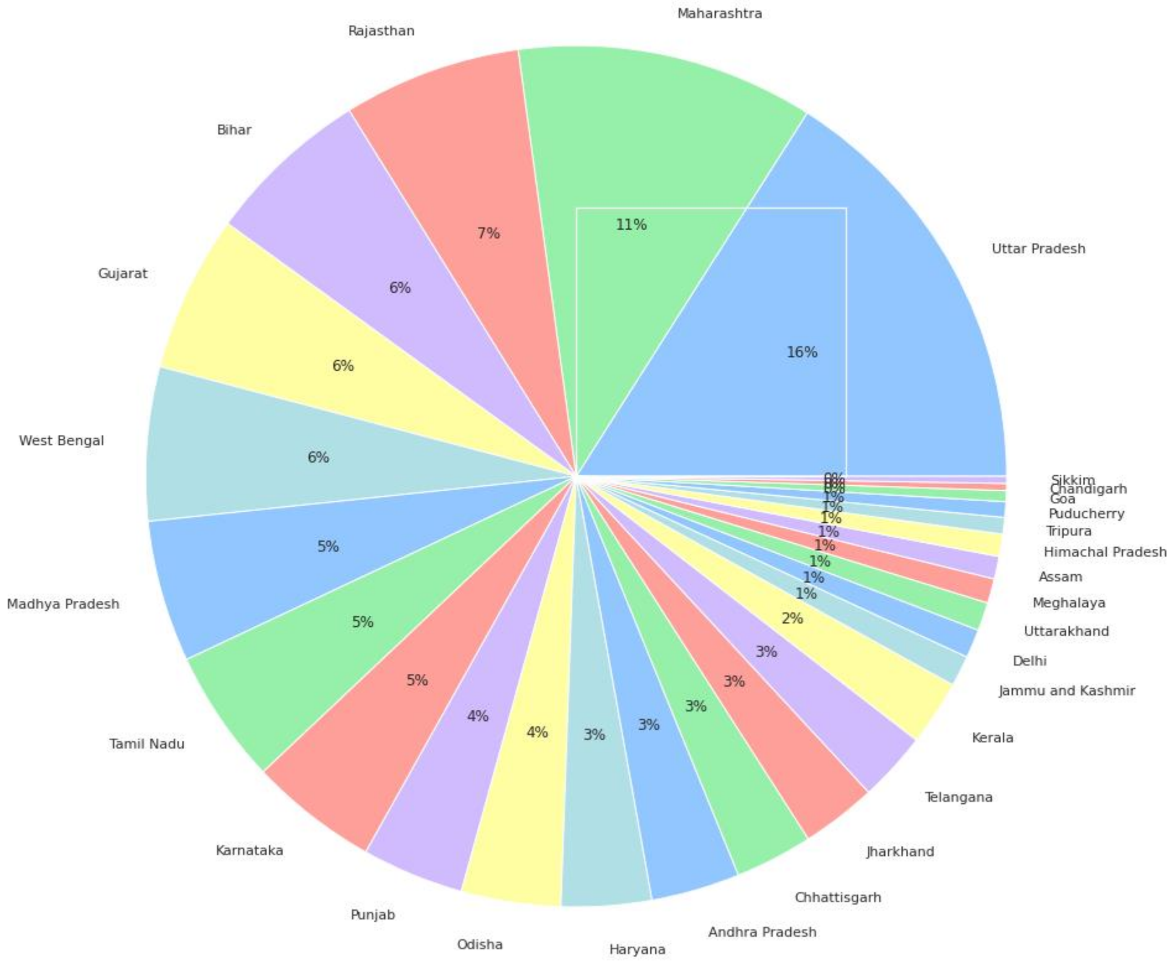
Visualization –

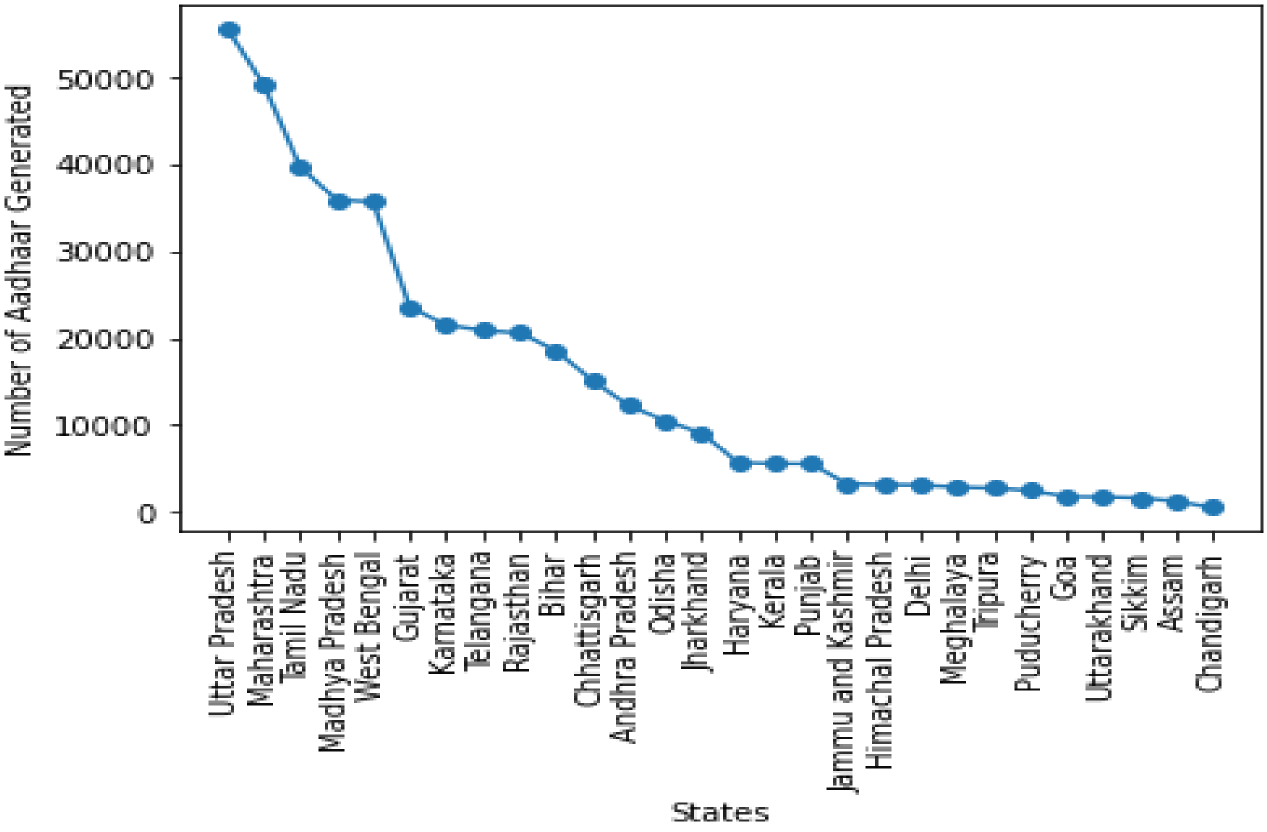


```
state=[]
val=[]
for datas in rdd:
  a,b = datas.split(' - ')
  state.append(a)
  val.append(int(b))
```

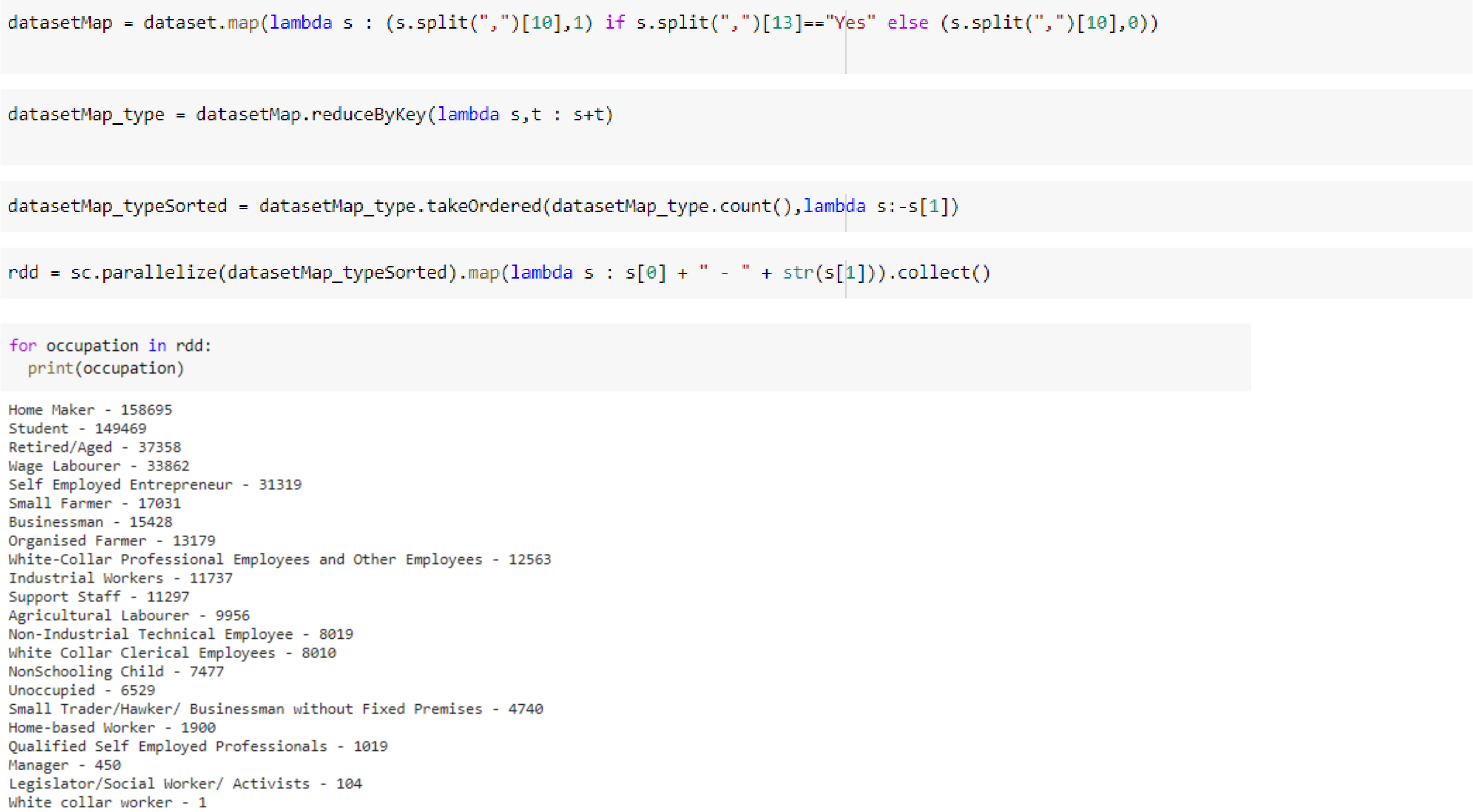


```
plt.figure(20.8)
plt.pie(val, labels = state, autopct='%.0f%%',radius=1.6, frame=True)
```


Maharashtra
Rajasthan
Bihar
Gujarat
West Bengal
Madhya Pradesh
Tamil Nadu
Karnataka
Punjab
Odisha
Haryana
Andhra Pradesh
Chhattisgarh
Jharkhand
Telangana
Kerala
Jammu and Kashmir
Delhi
Uttarakhand
Meghalaya
Assam
Himachal Pradesh
Tripura
Puducherry
Goa
Chandigarh
Sikkim
Uttar Pradesh
16%
11%
7%
6%
6%
6%
5%
5%
5%
4%
4%
3%
3%
3%
3%
3%
2%
1%
1%
1%
1%
1%
1%
1%
1%
0%
0%
0%


Number of Aadhaar Generated
50000
40000
30000
20000
10000
0
Uttar Pradesh
Maharashtra
Tamil Nadu
Madhya Pradesh
West Bengal
Gujarat
Karnataka
Telangana
Rajasthan
Bihar
Chhattisgarh
Andhra Pradesh
Odisha
Jharkhand
Haryana
Kerala
Punjab
Jammu and Kashmir
Himachal Pradesh
Delhi
Meghalaya
Tripura
Puducherry
Goa
Uttarakhand
Sikkim
Assam
Chandigarh
States

Count the people having Adhar Occupationwise –



```
datasetMap = dataset.map(lambda s : (s.split(",")[10],1) if s.split(",")[13]=="Yes" else (s.split(",")[10],0))
```

```
datasetMap_type = datasetMap.reduceByKey(lambda s,t : s+t)
```

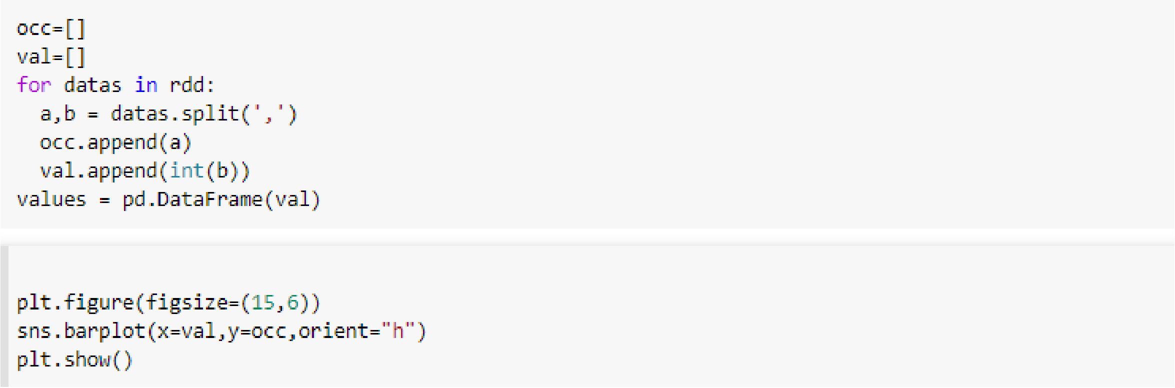
```
datasetMap_typeSorted = datasetMap_type.takeOrdered(datasetMap_type.count(),lambda s:-s[1])
```

```
rdd = sc.parallelize(datasetMap_typeSorted).map(lambda s : s[0] + " - " + str(s[1])).collect()
```

```
for occupation in rdd:
  print(occupation)
```

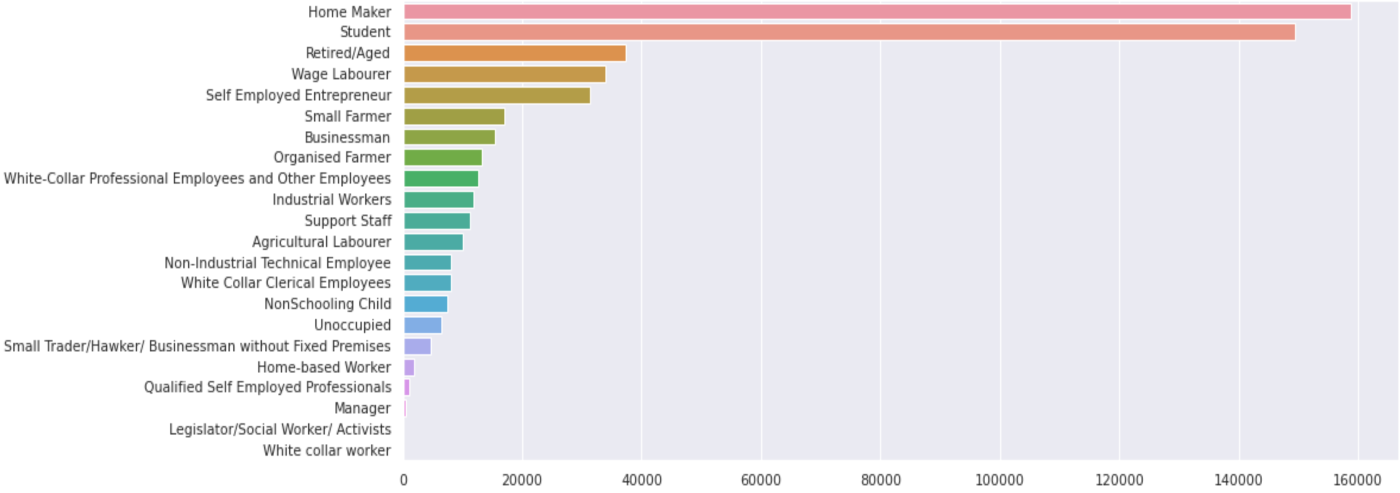
```
Home Maker - 158695
Student - 149469
Retired/Aged - 37358
Wage Labourer - 33862
Self Employed Entrepreneur - 31319
Small Farmer - 17031
Businessman - 15428
Organised Farmer - 13179
White-Collar Professional Employees and Other Employees - 12563
Industrial Workers - 11737
Support Staff - 11297
Agricultural Labourer - 9956
Non-Industrial Technical Employee - 8019
White Collar Clerical Employees - 8010
NonSchooling Child - 7477
Unoccupied - 6529
Small Trader/Hawker/ Businessman without Fixed Premises - 4740
Home-based Worker - 1900
Qualified Self Employed Professionals - 1019
Manager - 450
Legislator/Social Worker/ Activists - 104
White collar worker - 1
```

Visualizing –

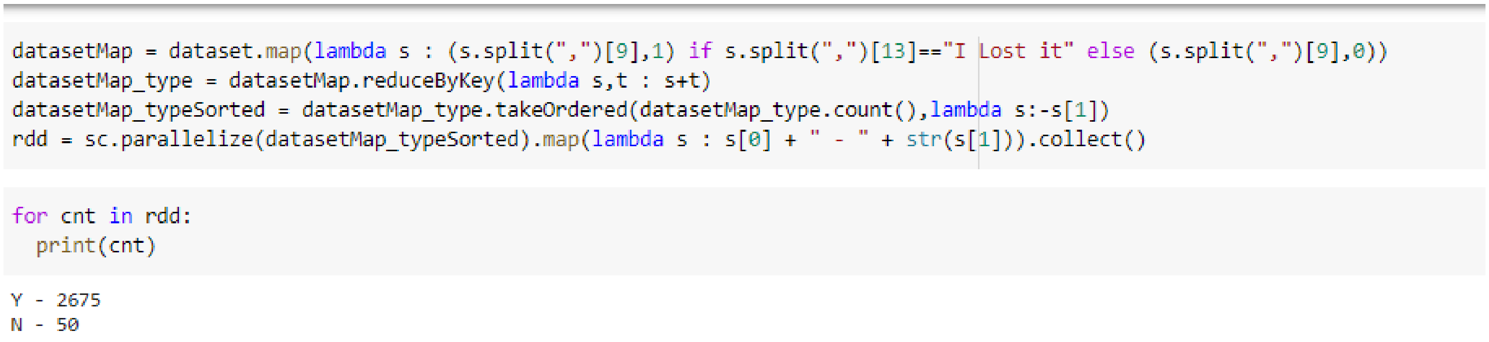


```
occ=[]
val=[]
for datas in rdd:
  a,b = datas.split(',')
  occ.append(a)
  val.append(int(b))
values = pd.DataFrame(val)
```

```
plt.figure(figsize=(15,6))
sns.barplot(x=val,y=occ,orient="h")
plt.show()
```



Count number of literate people who lost their adhar card –

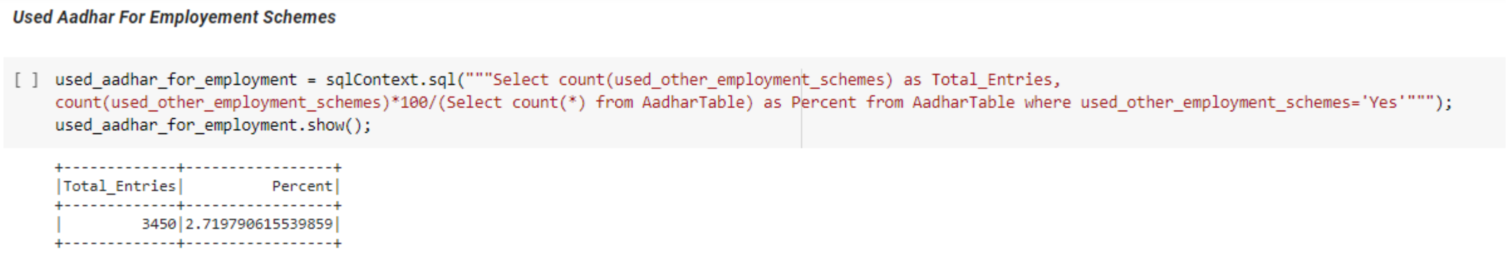


```
datasetMap = dataset.map(lambda s : (s.split(",")[9],1) if s.split(",")[13]=="I Lost it" else (s.split(",")[9],0))
datasetMap_type = datasetMap.reduceByKey(lambda s,t : s+t)
datasetMap_typeSorted = datasetMap_type.takeOrdered(datasetMap_type.count(),lambda s:-s[1])
rdd = sc.parallelize(datasetMap_typeSorted).map(lambda s : s[0] + " - " + str(s[1])).collect()
```

```
for cnt in rdd:
  print(cnt)
```

```
Y - 2675
N - 50
```

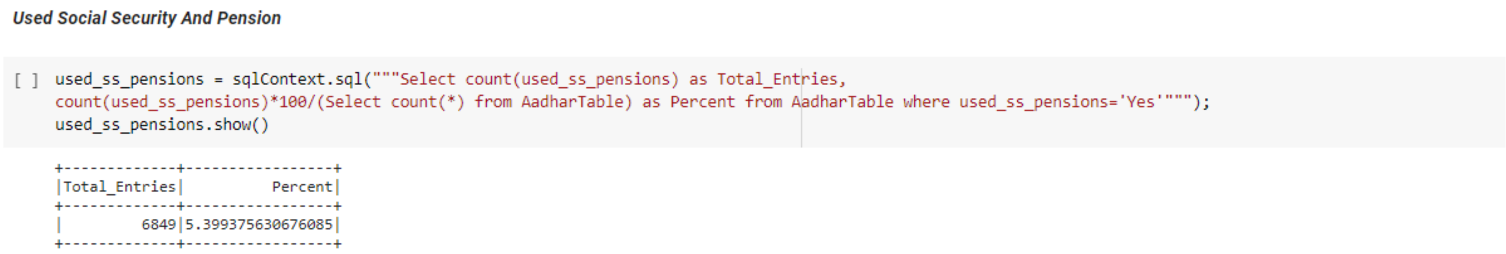
How People are using Adhar –



Used Aadhar For Employement Schemes

```
[ ] used_aadhar_for_employment = sqlContext.sql("""Select count(used_other_employment_schemes) as Total_Entries,
    count(used_other_employment_schemes)*100/(Select count(*) from AadharTable) as Percent from AadharTable where used_other_employment_schemes='Yes'""");
    used_aadhar_for_employment.show();
```

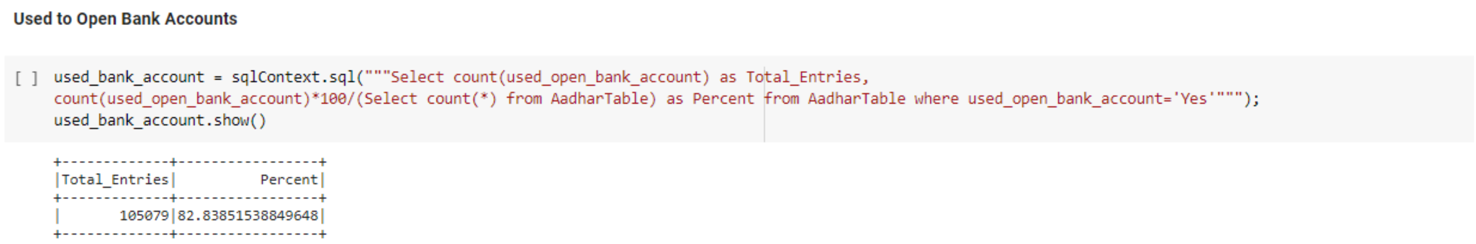
```
+-------------+-----------------+
|Total_Entries|          Percent|
+-------------+-----------------+
|         3450|2.719790615539859|
+-------------+-----------------+
```



Used Social Security And Pension

```
[ ] used_ss_pensions = sqlContext.sql("""Select count(used_ss_pensions) as Total_Entries,
    count(used_ss_pensions)*100/(Select count(*) from AadharTable) as Percent from AadharTable where used_ss_pensions='Yes'""");
    used_ss_pensions.show()
```

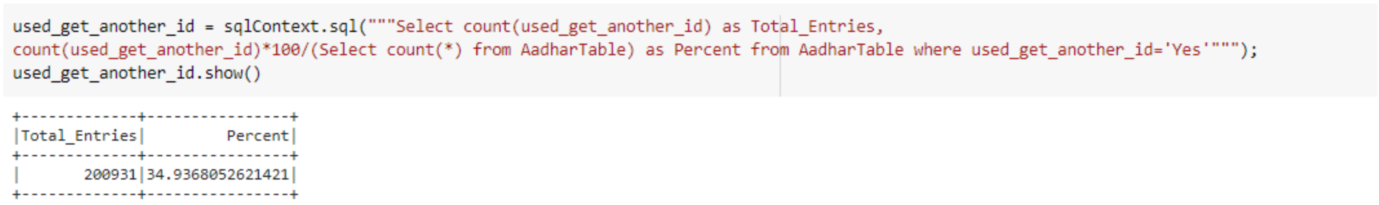
```
+-------------+-----------------+
|Total_Entries|          Percent|
+-------------+-----------------+
|         6849|5.399375630676085|
+-------------+-----------------+
```



Used to Open Bank Accounts

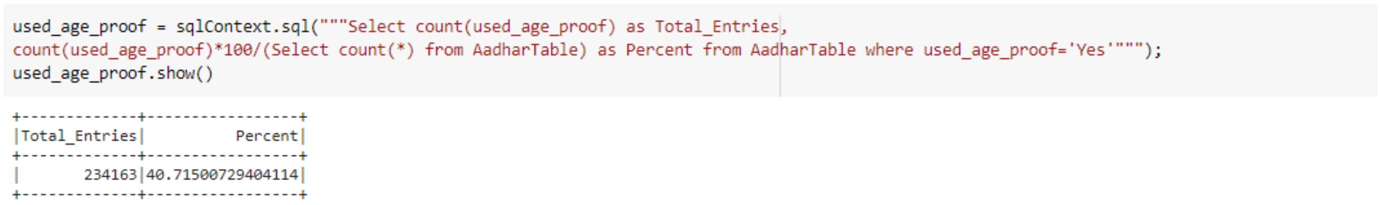
```
[ ] used_bank_account = sqlContext.sql("""Select count(used_open_bank_account) as Total_Entries,
    count(used_open_bank_account)*100/(Select count(*) from AadharTable) as Percent from AadharTable where used_open_bank_account='Yes'""");
    used_bank_account.show()
```

```
+-------------+-----------------+
|Total_Entries|          Percent|
+-------------+-----------------+
|       105079|82.83851538849648|
+-------------+-----------------+
```



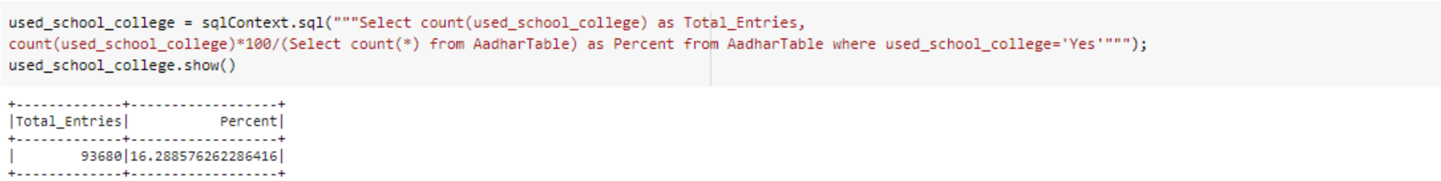
```
used_get_another_id = sqlContext.sql("""Select count(used_get_another_id) as Total_Entries,
count(used_get_another_id)*100/(Select count(*) from AadharTable) as Percent from AadharTable where used_get_another_id='Yes'""");
used_get_another_id.show()
```

```
+-------------+----------------+
|Total_Entries|         Percent|
+-------------+----------------+
|       200931|34.9368052621421|
+-------------+----------------+
```



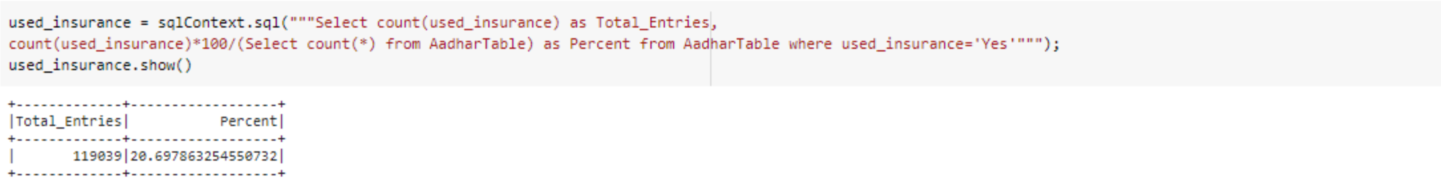
```
used_age_proof = sqlContext.sql("""Select count(used_age_proof) as Total_Entries,
count(used_age_proof)*100/(Select count(*) from AadharTable) as Percent from AadharTable where used_age_proof='Yes'""");
used_age_proof.show()
```

```
+-------------+-----------------+
|Total_Entries|          Percent|
+-------------+-----------------+
|       234163|40.71500729404114|
+-------------+-----------------+
```



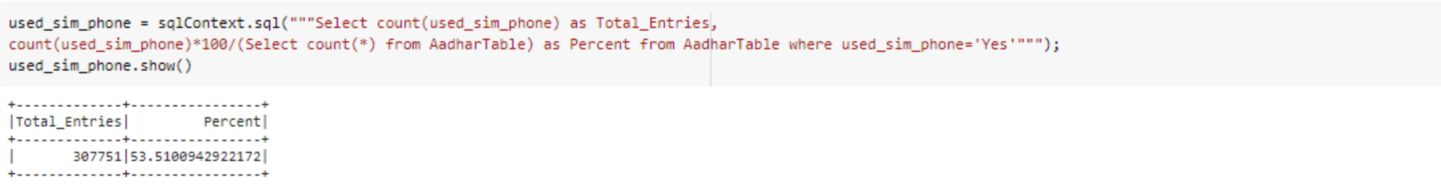
```
used_school_college = sqlContext.sql("""Select count(used_school_college) as Total_Entries,
count(used_school_college)*100/(Select count(*) from AadharTable) as Percent from AadharTable where used_school_college='Yes'""");
used_school_college.show()
```

```
+-------------+-----------------+
|Total_Entries|          Percent|
+-------------+-----------------+
|        93680|16.288576262286416|
+-------------+-----------------+
```



```
used_insurance = sqlContext.sql("""Select count(used_insurance) as Total_Entries,
count(used_insurance)*100/(Select count(*) from AadharTable) as Percent from AadharTable where used_insurance='Yes'""");
used_insurance.show()
```

```
+-------------+-----------------+
|Total_Entries|          Percent|
+-------------+-----------------+
|       119039|20.697863254550732|
+-------------+-----------------+
```

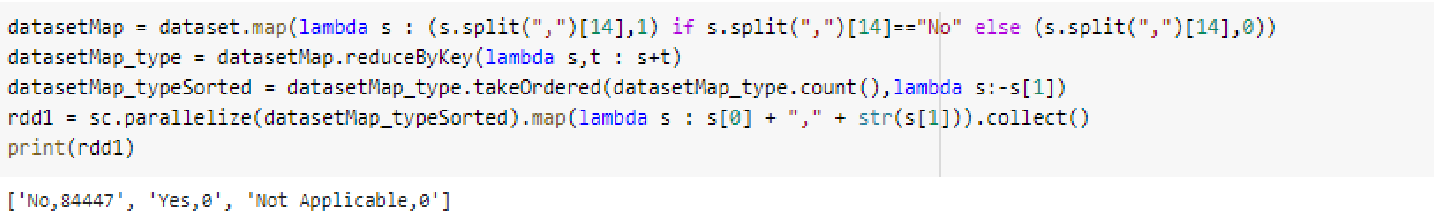


```
used_sim_phone = sqlContext.sql("""Select count(used_sim_phone) as Total_Entries,
count(used_sim_phone)*100/(Select count(*) from AadharTable) as Percent from AadharTable where used_sim_phone='Yes'""");
used_sim_phone.show()
```

```
+-------------+---------------+
|Total_Entries|        Percent|
+-------------+---------------+
|       307751|53.5100942922172|
+-------------+---------------+
```

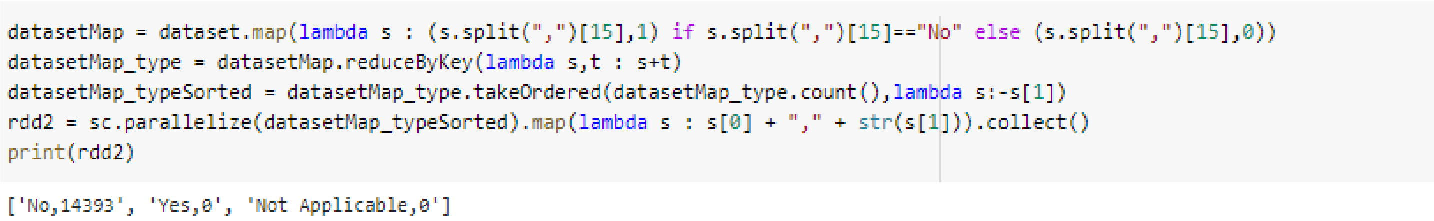
We can observe from the above questions that most people utilise their adhar card to buy new SIM cards and open bank accounts. As well as being age-proof.

Incorrect Adhar data Analysis –



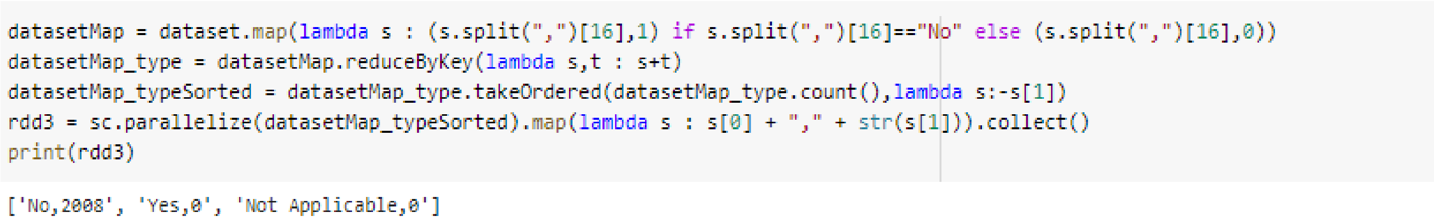
```
datasetMap = dataset.map(lambda s : (s.split(",")[14],1) if s.split(",")[14]=="No" else (s.split(",")[14],0))
datasetMap_type = datasetMap.reduceByKey(lambda s,t : s+t)
datasetMap_typeSorted = datasetMap_type.takeOrdered(datasetMap_type.count(),lambda s:-s[1])
rdd1 = sc.parallelize(datasetMap_typeSorted).map(lambda s : s[0] + "," + str(s[1])).collect()
print(rdd1)
```

```
['No,84447', 'Yes,0', 'Not Applicable,0']
```



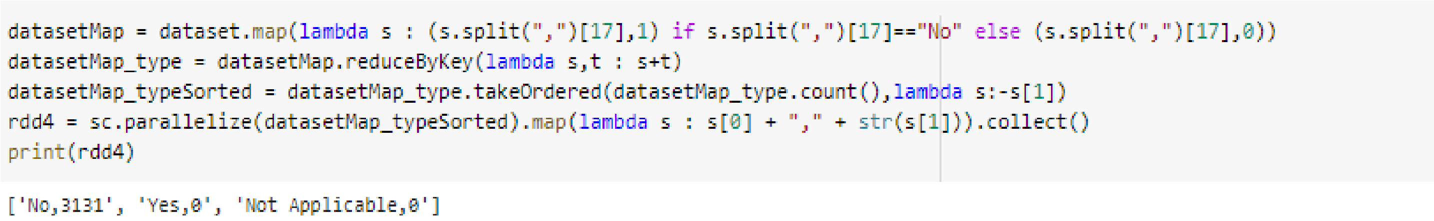
```
datasetMap = dataset.map(lambda s : (s.split(",")[15],1) if s.split(",")[15]=="No" else (s.split(",")[15],0))
datasetMap_type = datasetMap.reduceByKey(lambda s,t : s+t)
datasetMap_typeSorted = datasetMap_type.takeOrdered(datasetMap_type.count(),lambda s:-s[1])
rdd2 = sc.parallelize(datasetMap_typeSorted).map(lambda s : s[0] + "," + str(s[1])).collect()
print(rdd2)
```

```
['No,14393', 'Yes,0', 'Not Applicable,0']
```



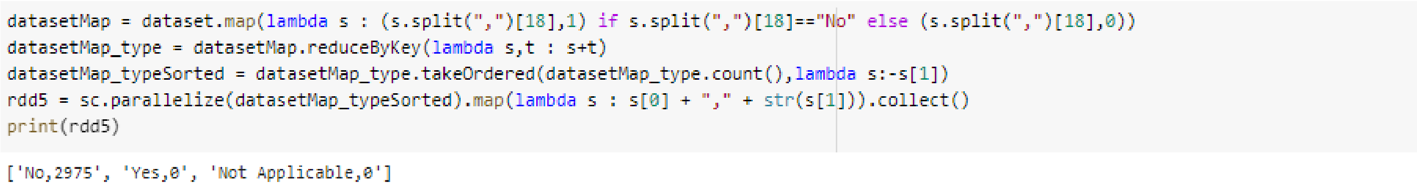
```
datasetMap = dataset.map(lambda s : (s.split(",")[16],1) if s.split(",")[16]=="No" else (s.split(",")[16],0))
datasetMap_type = datasetMap.reduceByKey(lambda s,t : s+t)
datasetMap_typeSorted = datasetMap_type.takeOrdered(datasetMap_type.count(),lambda s:-s[1])
rdd3 = sc.parallelize(datasetMap_typeSorted).map(lambda s : s[0] + "," + str(s[1])).collect()
print(rdd3)
```

```
['No,2008', 'Yes,0', 'Not Applicable,0']
```



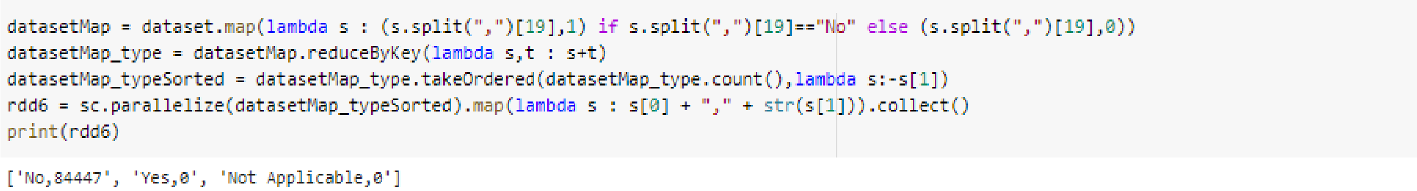
```
datasetMap = dataset.map(lambda s : (s.split(",")[17],1) if s.split(",")[17]=="No" else (s.split(",")[17],0))
datasetMap_type = datasetMap.reduceByKey(lambda s,t : s+t)
datasetMap_typeSorted = datasetMap_type.takeOrdered(datasetMap_type.count(),lambda s:-s[1])
rdd4 = sc.parallelize(datasetMap_typeSorted).map(lambda s : s[0] + "," + str(s[1])).collect()
print(rdd4)
```

```
['No,3131', 'Yes,0', 'Not Applicable,0']
```



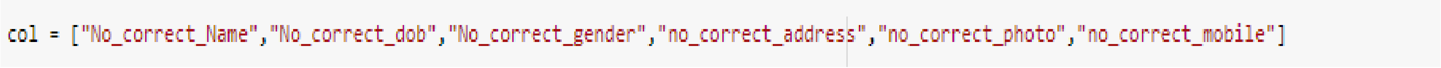
```
datasetMap = dataset.map(lambda s : (s.split(",")[18],1) if s.split(",")[18]=="No" else (s.split(",")[18],0))
datasetMap_type = datasetMap.reduceByKey(lambda s,t : s+t)
datasetMap_typeSorted = datasetMap_type.takeOrdered(datasetMap_type.count(),lambda s:-s[1])
rdd5 = sc.parallelize(datasetMap_typeSorted).map(lambda s : s[0] + "," + str(s[1])).collect()
print(rdd5)
```

```
['No,2975', 'Yes,0', 'Not Applicable,0']
```

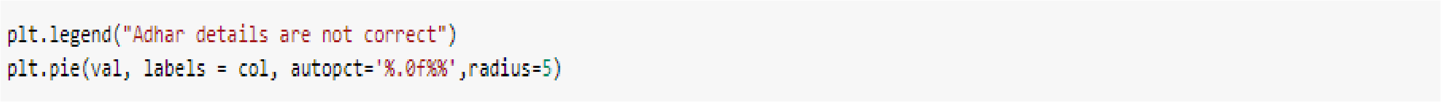


```
datasetMap = dataset.map(lambda s : (s.split(",")[19],1) if s.split(",")[19]=="No" else (s.split(",")[19],0))
datasetMap_type = datasetMap.reduceByKey(lambda s,t : s+t)
datasetMap_typeSorted = datasetMap_type.takeOrdered(datasetMap_type.count(),lambda s:-s[1])
rdd6 = sc.parallelize(datasetMap_typeSorted).map(lambda s : s[0] + "," + str(s[1])).collect()
print(rdd6)
```

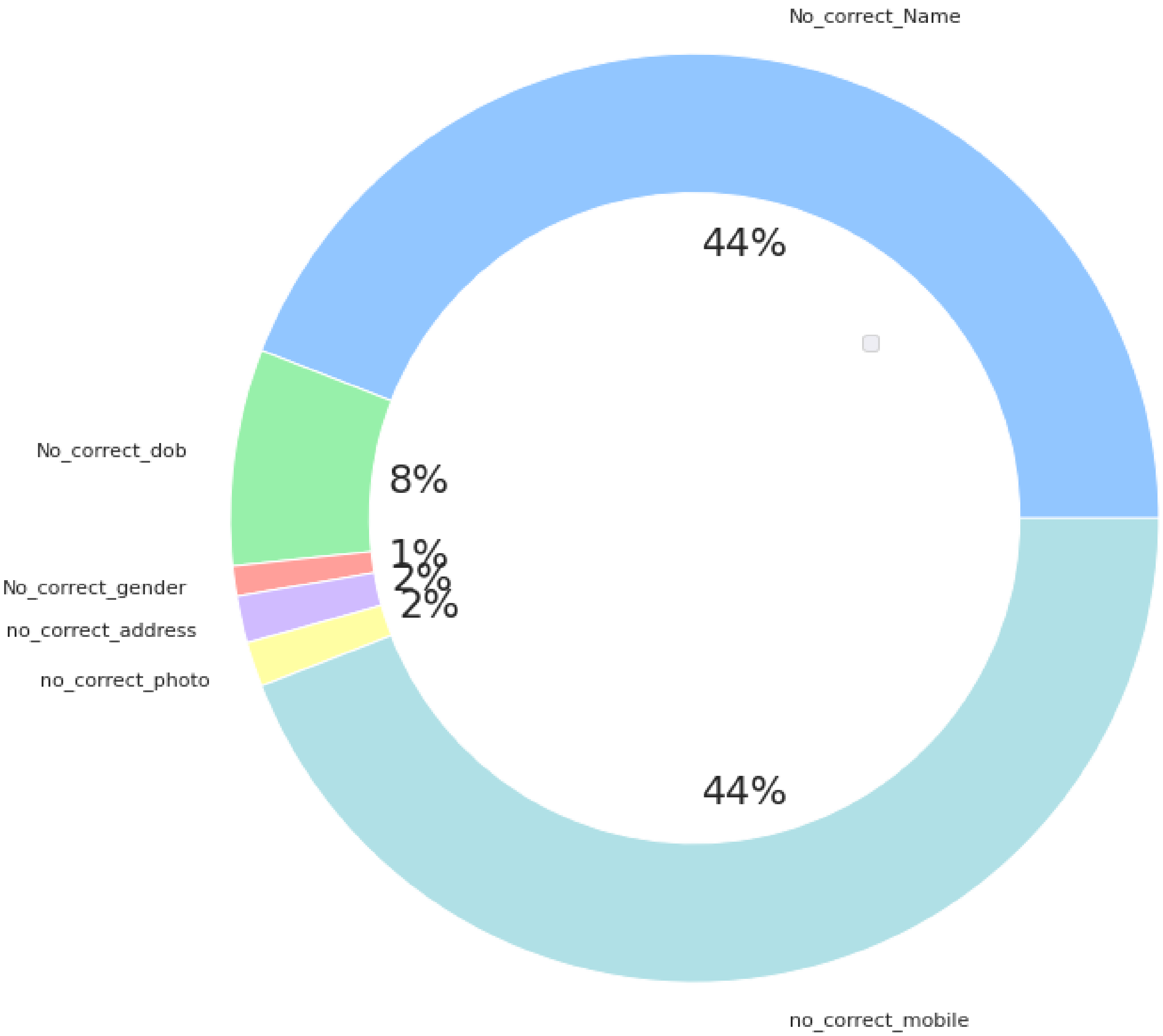
```
['No,84447', 'Yes,0', 'Not Applicable,0']
```



```
col = ["No_correct_Name","No_correct_dob","No_correct_gender","no_correct_address","no_correct_photo","no_correct_mobile"]
```

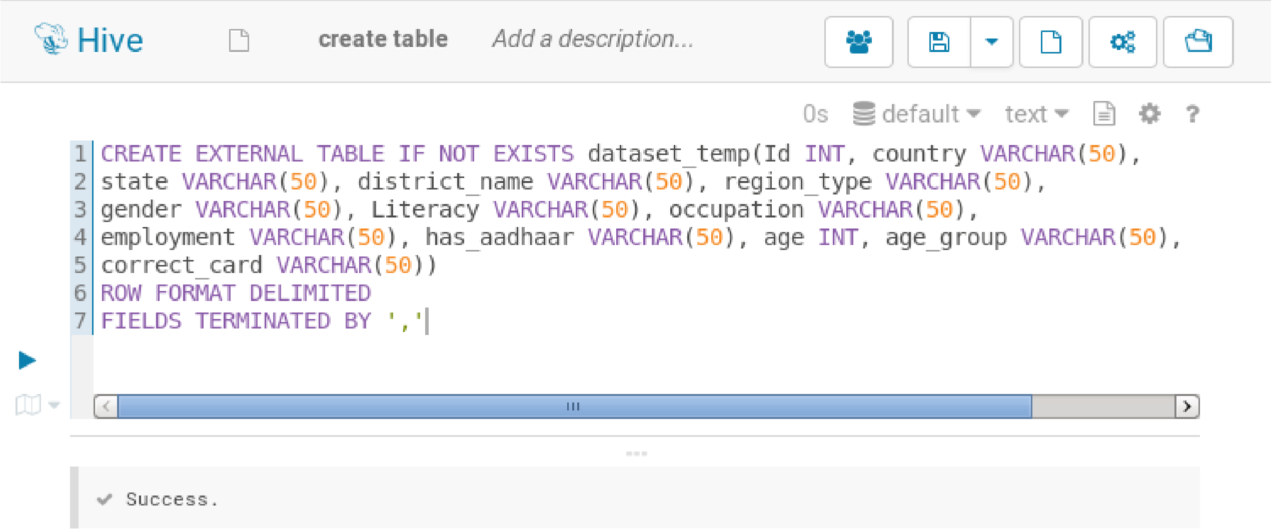


```
plt.legend("Adhar details are not correct")
plt.pie(val, labels = col, autopct='%.0f%%',radius=5)
```



Loading data into Hive –

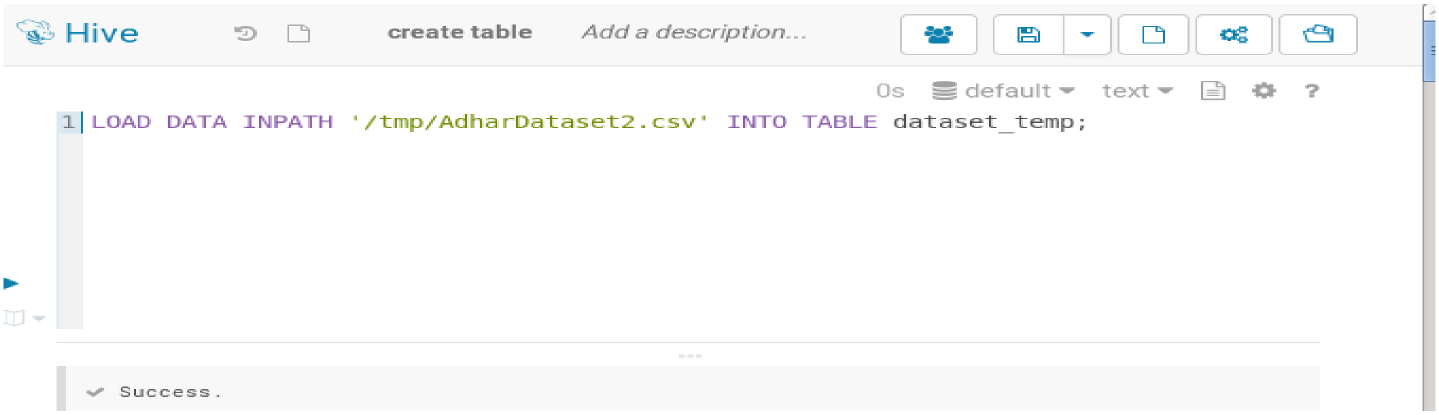
New table created and loaded in Hive –



```
CREATE EXTERNAL TABLE IF NOT EXISTS dataset_temp(Id INT, country VARCHAR(50),
state VARCHAR(50), district_name VARCHAR(50), region_type VARCHAR(50),
gender VARCHAR(50), Literacy VARCHAR(50), occupation VARCHAR(50),
employment VARCHAR(50), has_aadhaar VARCHAR(50), age INT, age_group VARCHAR(50),
correct_card VARCHAR(50))
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
```

```
✓ Success.
```

Loading data into table –

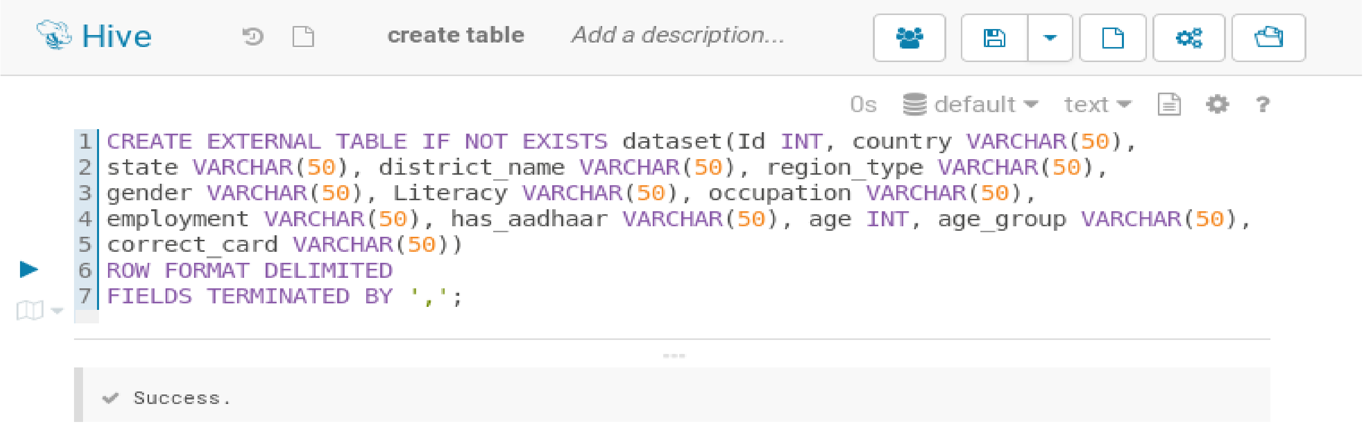


```
LOAD DATA INPATH '/tmp/AdharDataset2.csv' INTO TABLE dataset_temp;
```

```
✓ Success.
```

Removing header –

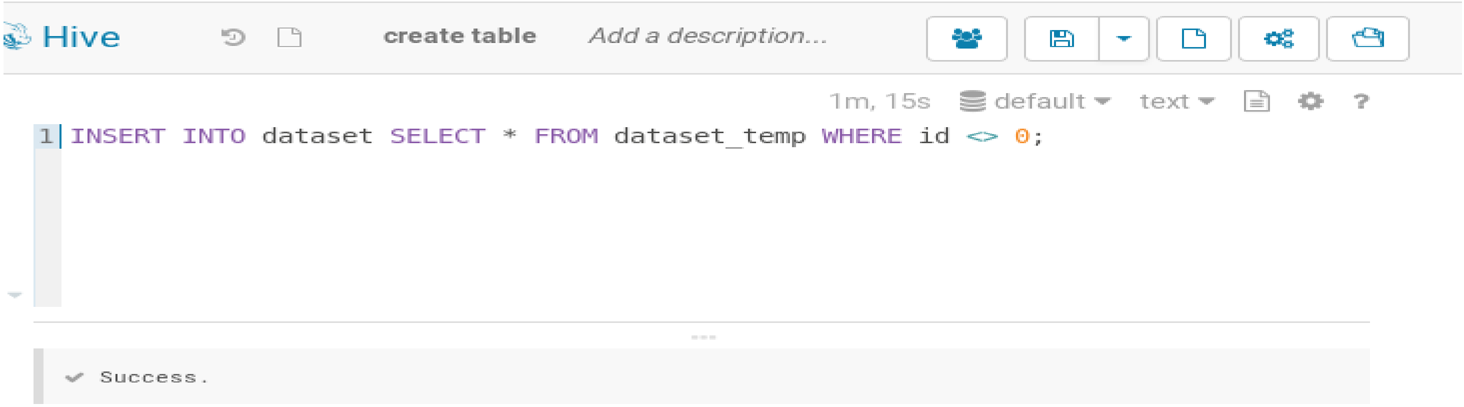
Actual table creation –



```
CREATE EXTERNAL TABLE IF NOT EXISTS dataset(Id INT, country VARCHAR(50),
state VARCHAR(50), district_name VARCHAR(50), region_type VARCHAR(50),
gender VARCHAR(50), Literacy VARCHAR(50), occupation VARCHAR(50),
employment VARCHAR(50), has_aadhaar VARCHAR(50), age INT, age_group VARCHAR(50),
correct_card VARCHAR(50))
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';
```

```
✓ Success.
```

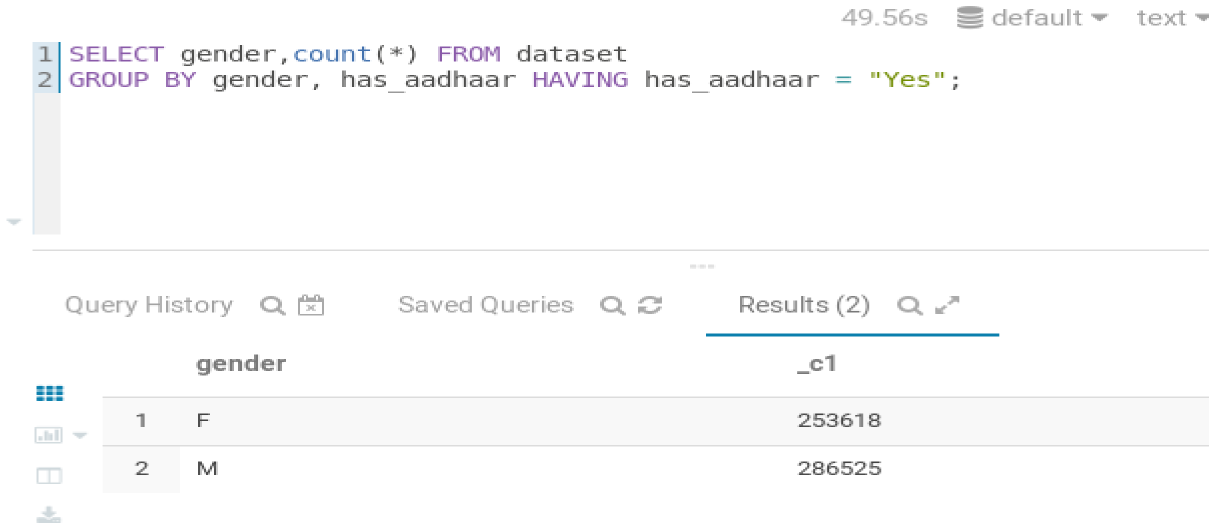
Inserting into actual table from temp table –



```
INSERT INTO dataset SELECT * FROM dataset_temp WHERE id <> 0;
```

```
✓ Success.
```

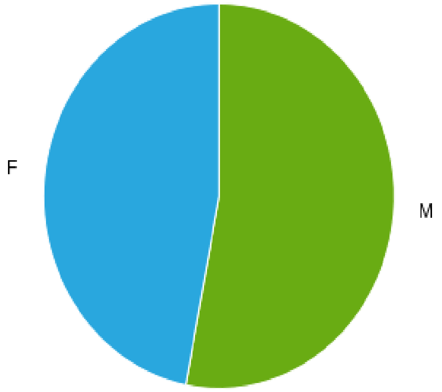
Gender wise Analysis – Having Adhar card

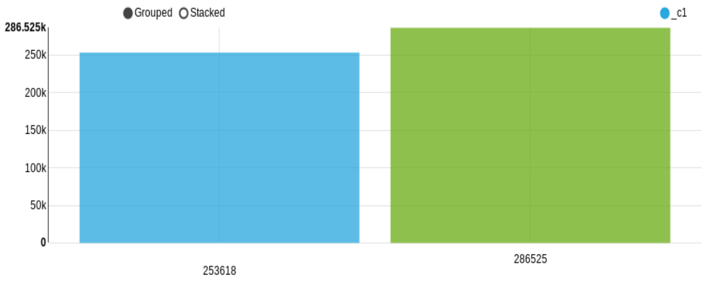


```
SELECT gender,count(*) FROM dataset
GROUP BY gender, has_aadhaar HAVING has_aadhaar = "Yes";
```

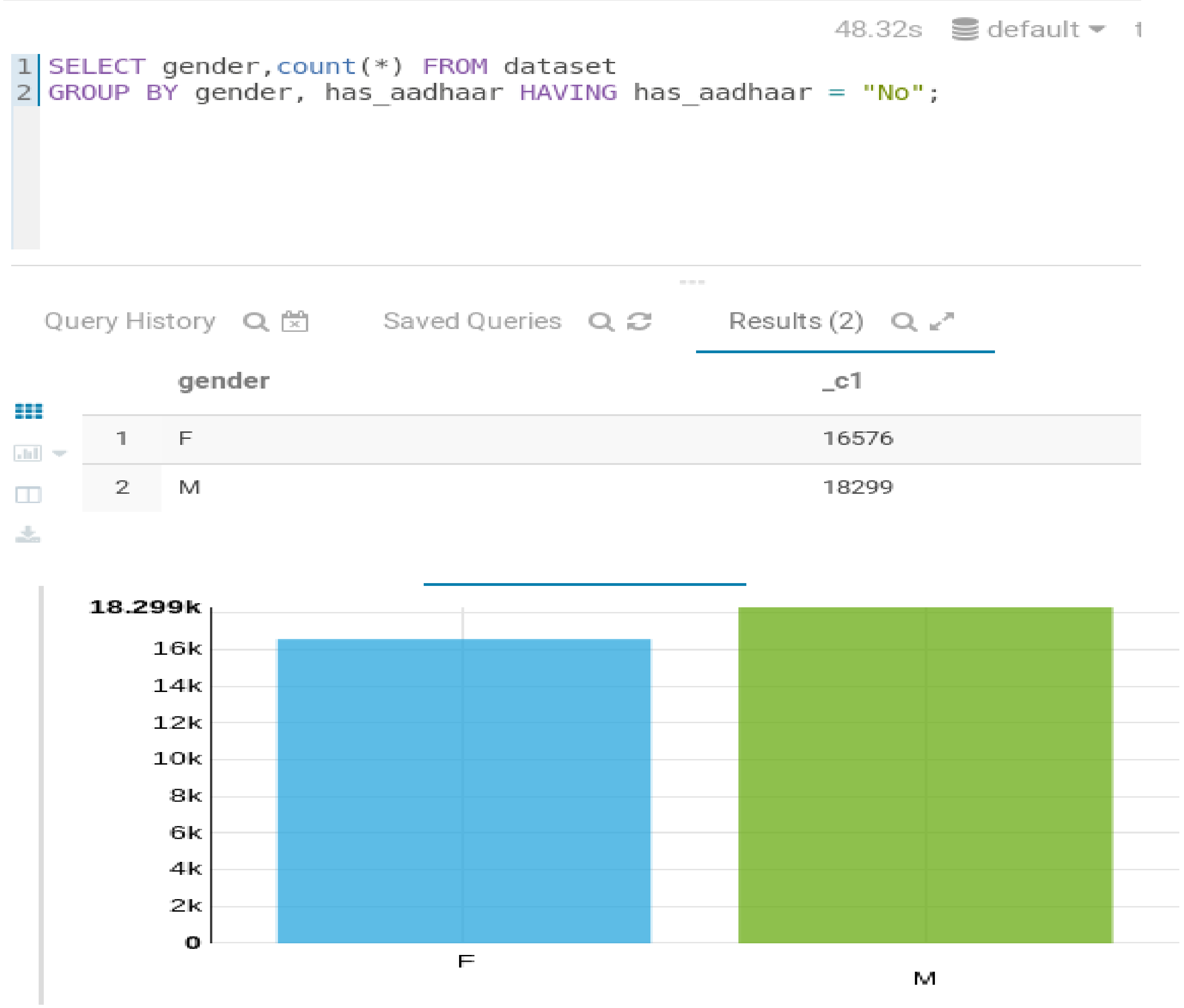
	gender	_c1
1	F	253618
2	M	286525

Visualization –





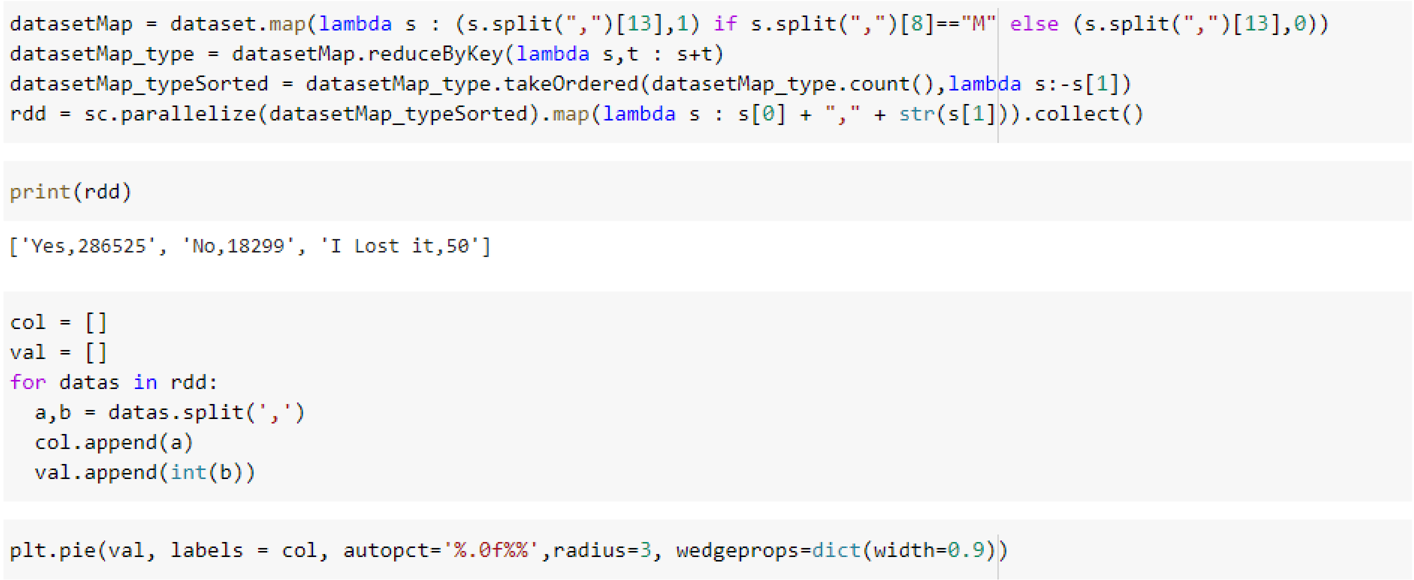
Not Having Adhar card –



```sql
SELECT gender,count(*) FROM dataset
GROUP BY gender, has_aadhaar HAVING has_aadhaar = "No";
```

	gender	_c1
1	F	16576
2	M	18299

Adhar details having gender = male



```python
datasetMap = dataset.map(lambda s : (s.split(",")[13],1) if s.split(",")[8]=="M" else (s.split(",")[13],0))
datasetMap_type = datasetMap.reduceByKey(lambda s,t : s+t)
datasetMap_typeSorted = datasetMap_type.takeOrdered(datasetMap_type.count(),lambda s:-s[1])
rdd = sc.parallelize(datasetMap_typeSorted).map(lambda s : s[0] + "," + str(s[1])).collect()
```

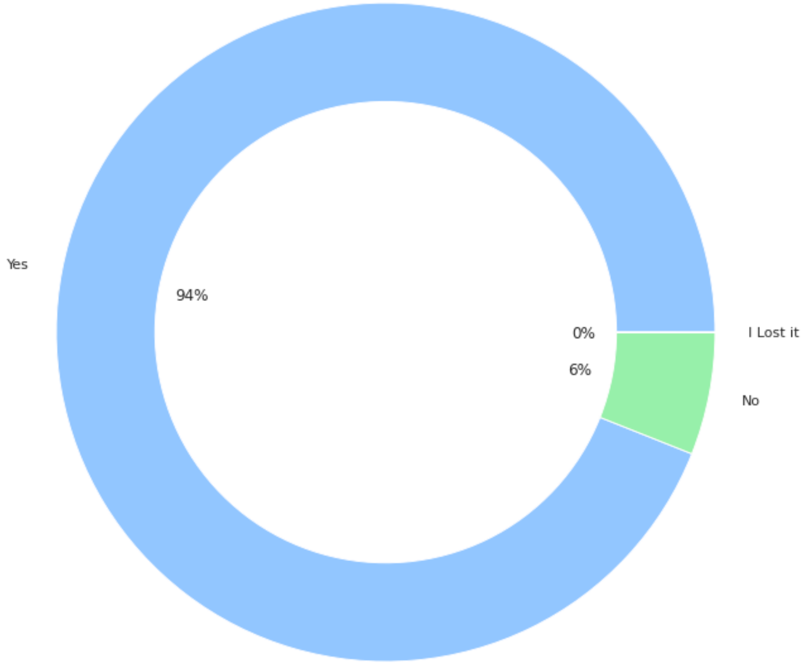
```python
print(rdd)
```

```
['Yes,286525', 'No,18299', 'I Lost it,50']
```

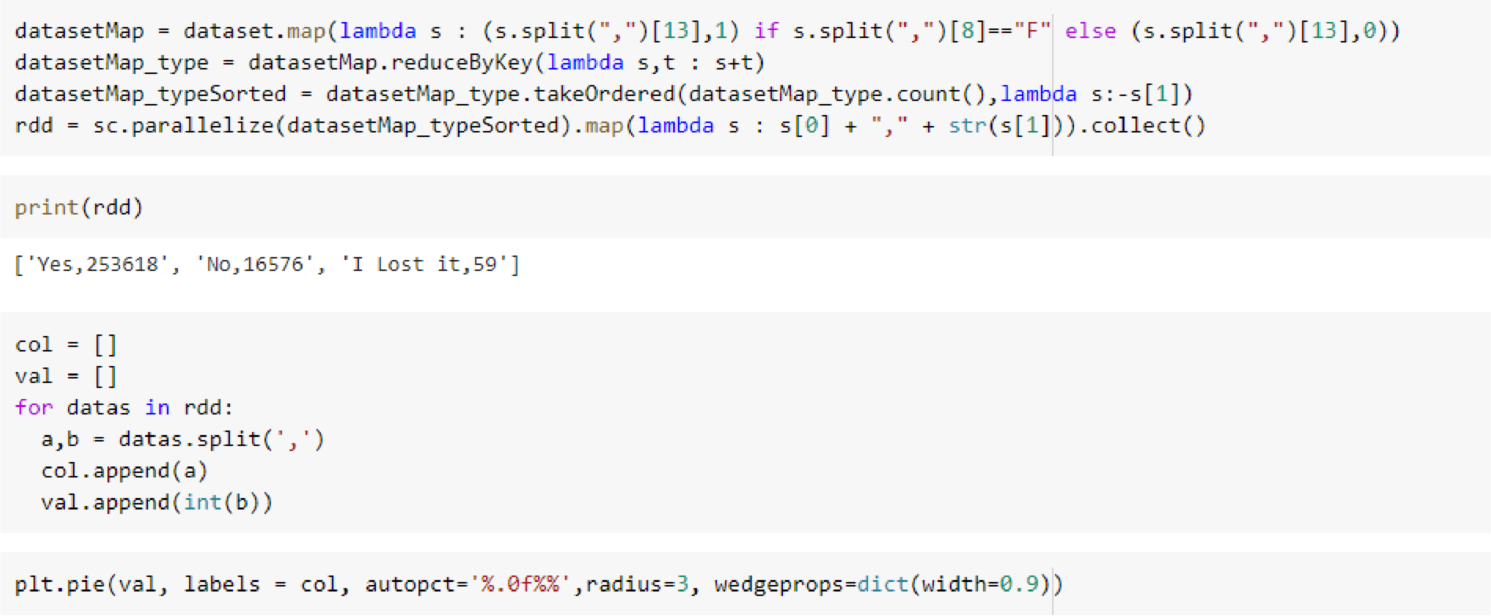
```python
col = []
val = []
for datas in rdd:
  a,b = datas.split(',')
  col.append(a)
  val.append(int(b))
```

```python
plt.pie(val, labels = col, autopct='%.0f%%',radius=3, wedgeprops=dict(width=0.9))
```

Visualization –



Adhar details having gender = female



```
datasetMap = dataset.map(lambda s : (s.split(",")[13],1) if s.split(",")[8]=="F" else (s.split(",")[13],0))
datasetMap_type = datasetMap.reduceByKey(lambda s,t : s+t)
datasetMap_typeSorted = datasetMap_type.takeOrdered(datasetMap_type.count(),lambda s:-s[1])
rdd = sc.parallelize(datasetMap_typeSorted).map(lambda s : s[0] + "," + str(s[1])).collect()
```

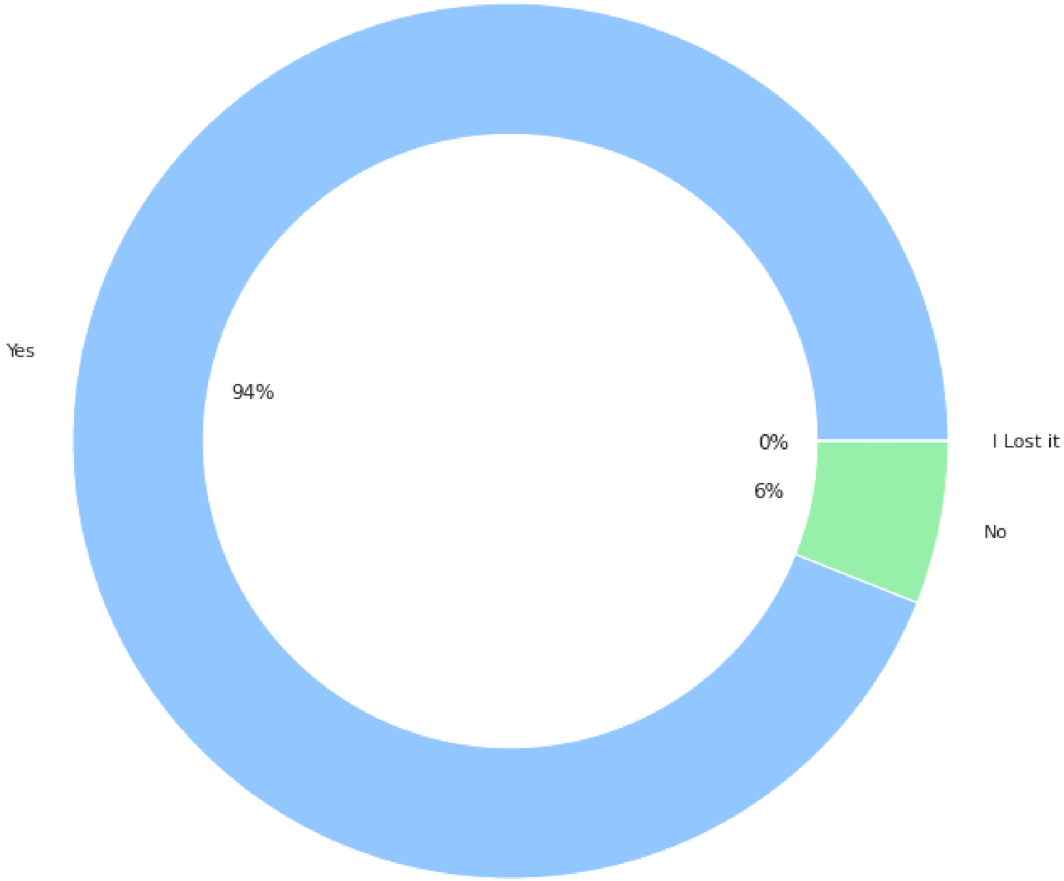
```
print(rdd)
```

```
['Yes,253618', 'No,16576', 'I Lost it,59']
```

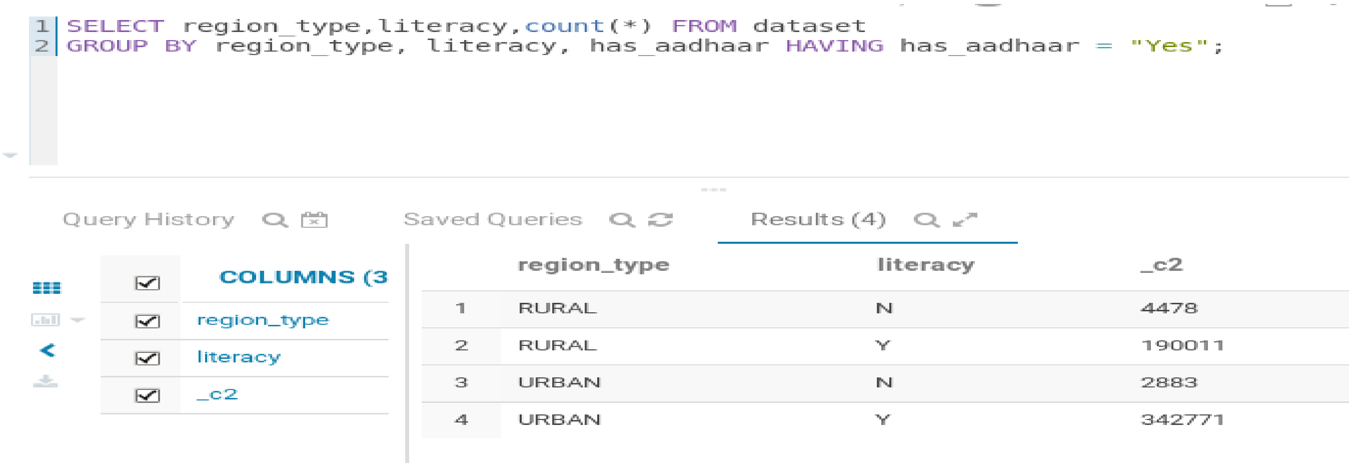
```
col = []
val = []
for datas in rdd:
  a,b = datas.split(',')
  col.append(a)
  val.append(int(b))
```

```
plt.pie(val, labels = col, autopct='%.0f%%',radius=3, wedgeprops=dict(width=0.9))
```

Visualization –



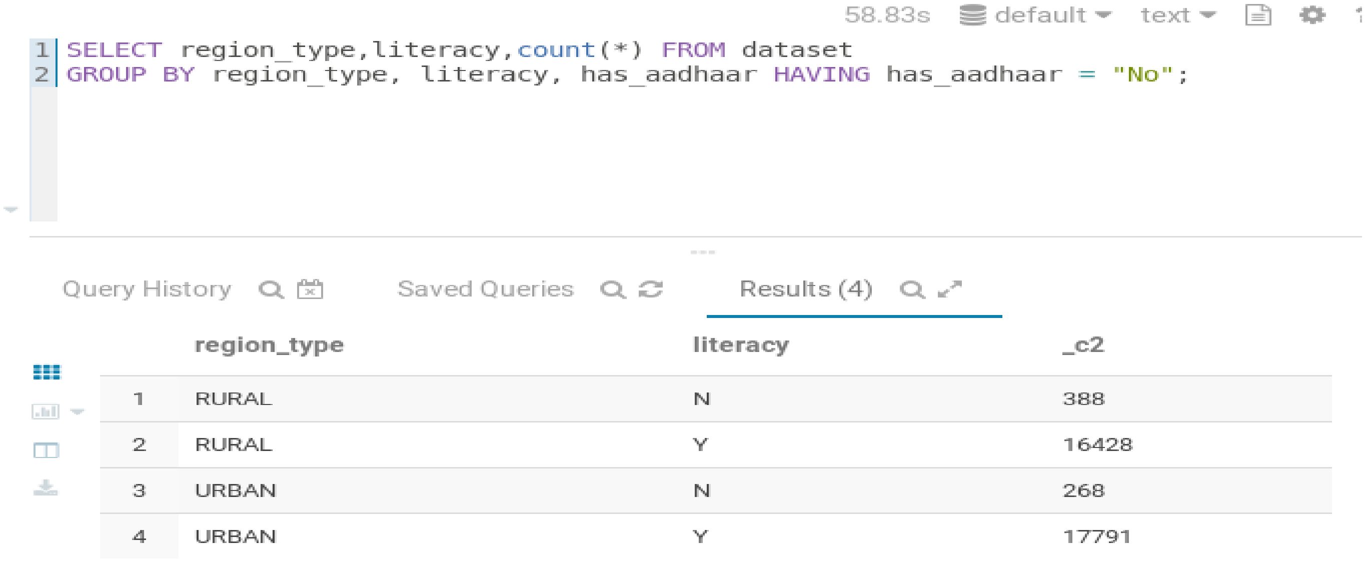
Data Analysis region type and literacy wise – (Having Adhar card)



```
SELECT region_type,literacy,count(*) FROM dataset
GROUP BY region_type, literacy, has_aadhaar HAVING has_aadhaar = "Yes";
```

	region_type	literacy	_c2
1	RURAL	N	4478
2	RURAL	Y	190011
3	URBAN	N	2883
4	URBAN	Y	342771

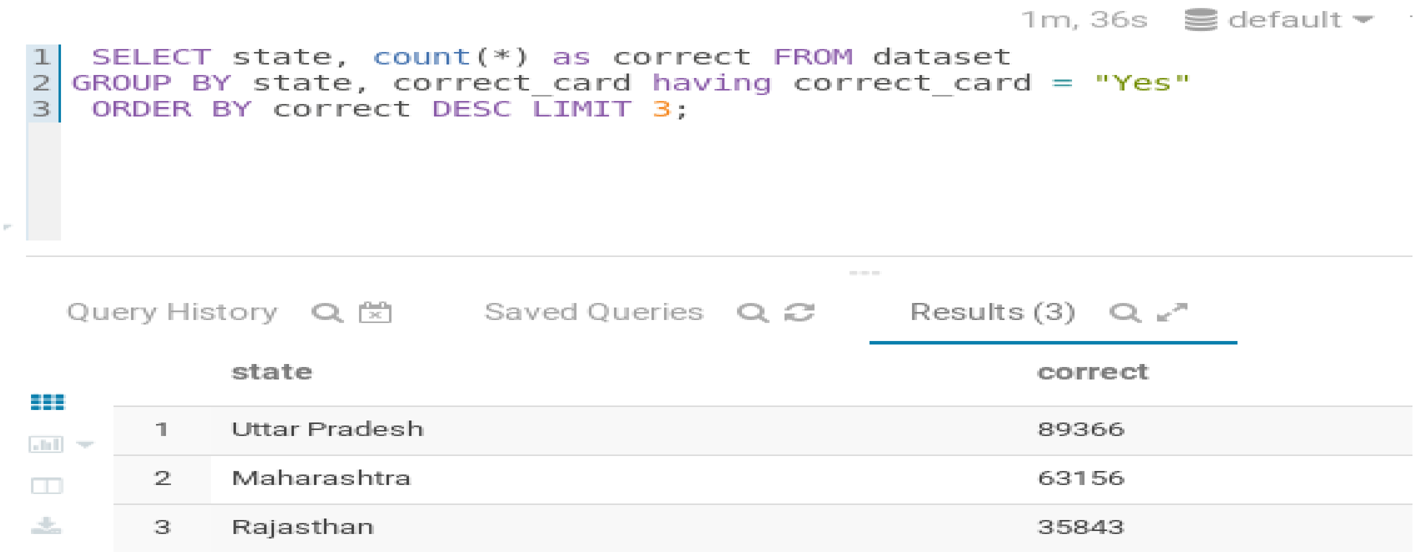
(Not having Adhar card) –



```
SELECT region_type,literacy,count(*) FROM dataset
GROUP BY region_type, literacy, has_aadhaar HAVING has_aadhaar = "No";
```

	region_type	literacy	_c2
1	RURAL	N	388
2	RURAL	Y	16428
3	URBAN	N	268
4	URBAN	Y	17791

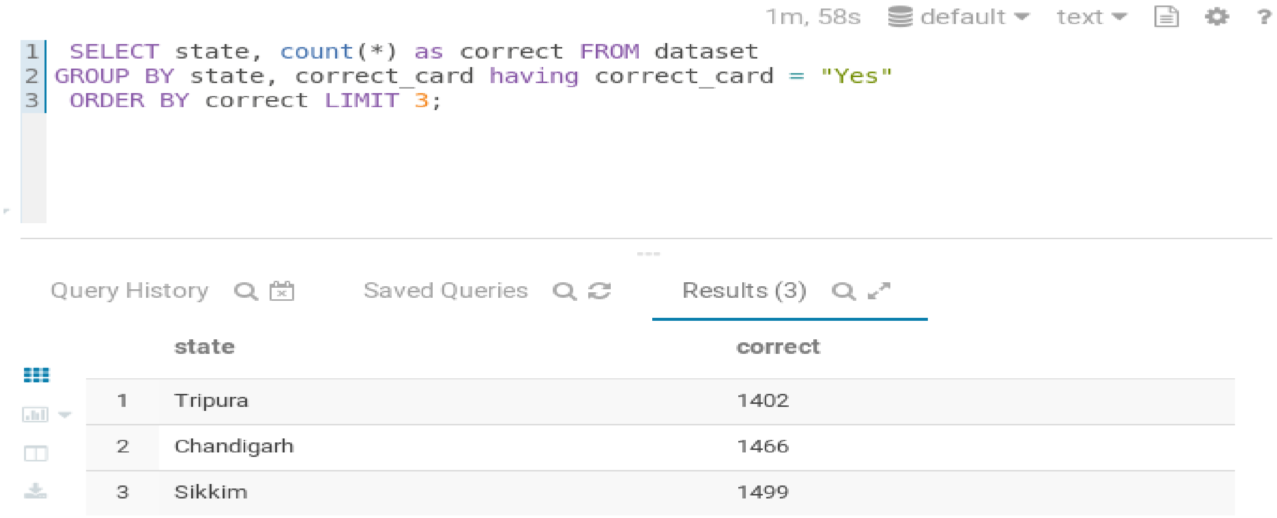
Top 3 states having Highest correct adhar cards



```
 SELECT state, count(*) as correct FROM dataset
GROUP BY state, correct_card having correct_card = "Yes"
 ORDER BY correct DESC LIMIT 3;
```

	state	correct
1	Uttar Pradesh	89366
2	Maharashtra	63156
3	Rajasthan	35843

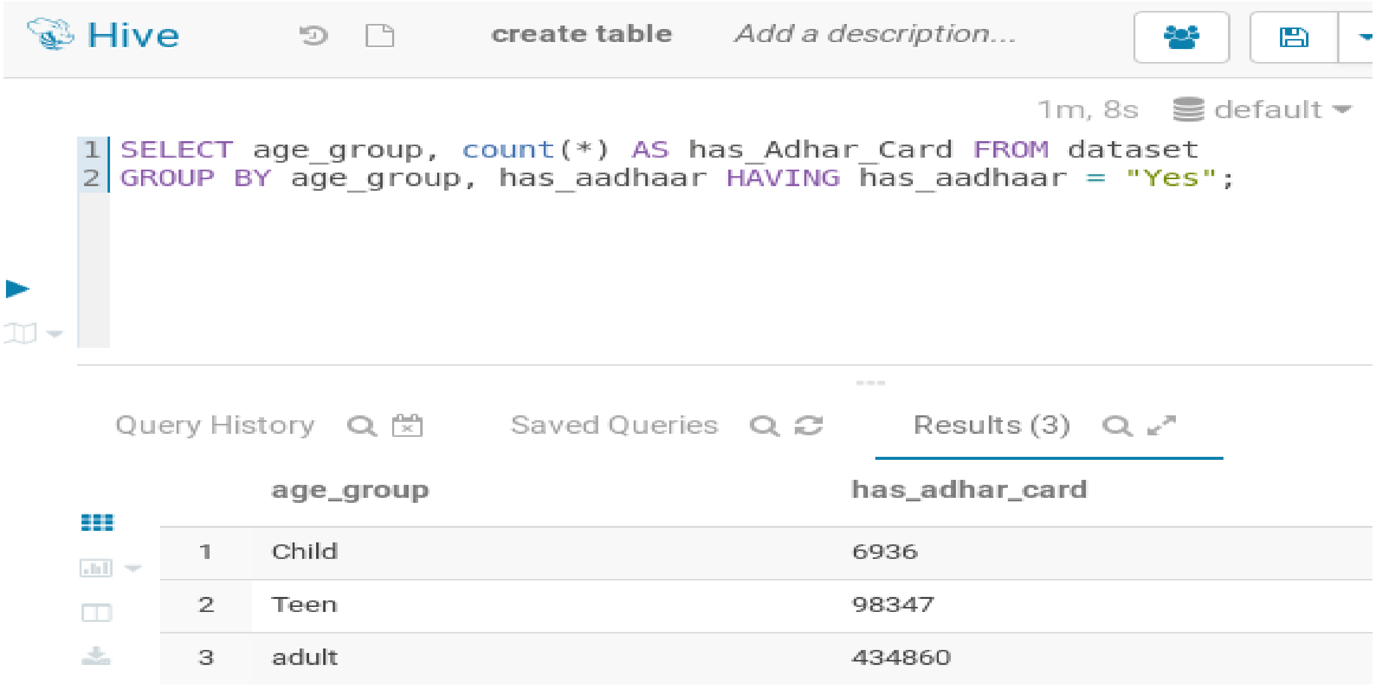
Top 3 states having Lowest correct adhar cards



```
 SELECT state, count(*) as correct FROM dataset
GROUP BY state, correct_card having correct_card = "Yes"
 ORDER BY correct LIMIT 3;
```

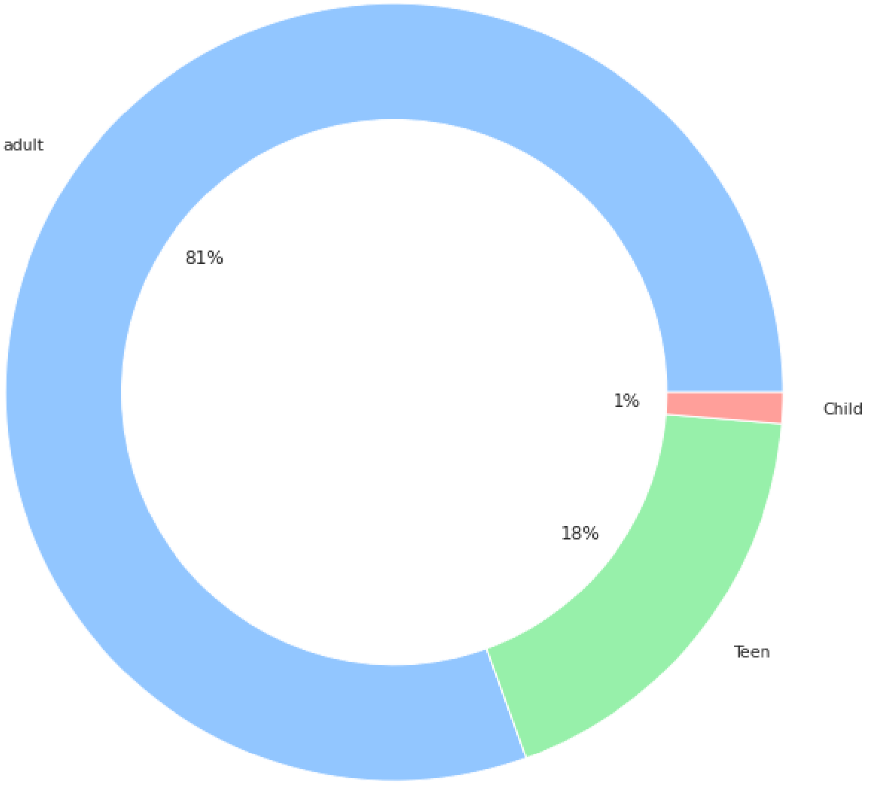
	state	correct
1	Tripura	1402
2	Chandigarh	1466
3	Sikkim	1499

Adhar data analysis according to age –



```
SELECT age_group, count(*) AS has_Adhar_Card FROM dataset
GROUP BY age_group, has_aadhaar HAVING has_aadhaar = "Yes";
```

	age_group	has_adhar_card
1	Child	6936
2	Teen	98347
3	adult	434860



6. RESULTS AND CONCLUSION

We hope our efforts build a shared understanding of the facts. The State of Aadhaar 2019 report highlights the most significant findings and themes across the study as well as an overview of the methodology and its limitations.

Residents expressed satisfaction with Aadhaar and trust in the system, despite ongoing difficulties, but emerging concerns need to be addressed.

Individuals who do not have Aadhaar or who face difficulties in using it are often those most in need of government support.

Making Aadhaar mandatory can lead to exclusion from welfare and other services—and place an additional burden on residents. Such mandates should therefore be carefully considered.

Variations in how states implement Aadhaar represent an opportunity to innovate and learn from each other’s successful practices.

Our ultimate aspiration is that policymakers, researchers, service providers, and others use the data and findings from the study to inform decisions about the future of Aadhaar and, more broadly, digital identity.

Limitations

Given the limitations of survey methodologies, we focused only on questions that residents were able to answer credibly through a survey format. There are many valuable questions, related to the experience of both people and providers that our study cannot answer—among them, the following:

To what extent can perceived benefits and challenges be attributed to Aadhaar?

What are residents’ worries about privacy and surveillance with respect to Aadhaar?

To what extent has Aadhaar benefitted the government?

To what extent has Aadhaar benefitted private-sector actors?

REFERENCES

[1] Pati, Rupesh & Kumar, Vipin & Jain, Nishtha. (2015). Analysis of Aadhaar: A Project Management Perspective. IIM Kozhikode Society & Management Review. 4. 124-135. 10.1177/2277975215610687.

[2] Pincha, Satyam & Lata, Kusum. (2018). Analytical Study on Aadhaar Card (UIDAI) and its inclusion into Public Services Delivery. 4. 64-68.

[3] Mohit Dayal, Nanhay Singh,An Anatomization of Aadhaar Card Data Set – A Big Data Challenge,Procedia Computer Science,Volume 85,2016,Pages 733-739,ISSN 1877-0509

[4] Pali, Isha & Krishania, Lisa & Chadha, Divya & Kandar, Asmita & Varshney, Gaurav & Shukla, Sneha. (2020). A Comprehensive Survey of Aadhar and Security Issues.

[5] Raju, Raja & Singh, Sukhdev & Khatter, Kiran. (2017). Aadhaar Card: Challenges and Impact on Digital Transformation.