HEALTH CARE SYSTEM ANALYSIS

Contents

1. Business Challenge / Requirement

A Health Care insurance company is facing challenges in enhancing its revenue and understanding the customers so it wants to take help of Big Data Ecosystem to analyze the Competitors company data received from varieties of sources, namely through scrapping and third-party sources. This analysis will help them to track the behavior, condition of customers so that to customize offers for them to buy insurance policies and also calculate royalties to those customers who buy policies in past, this in turn will enhance their revenues.

2. The goal of the project

The goal of the project is to create data pipelines for the Health Care insurance company which will make the company make appropriate business strategies to enhance their revenue by analyzing customers behaviors and send offers and royalties to customers respectively.

3. Data Flow Architecture / Process Flow

1. A Linux file server receives data files in form of json and csv. These files are coming from the third-party sources based on user interaction methodology.

2. The files data is validated, enriched and processed before loading into RDBMS System.

3. After validated the files data, we are creating the data model for RDBMS so that we can store the files data into the RDBMS.

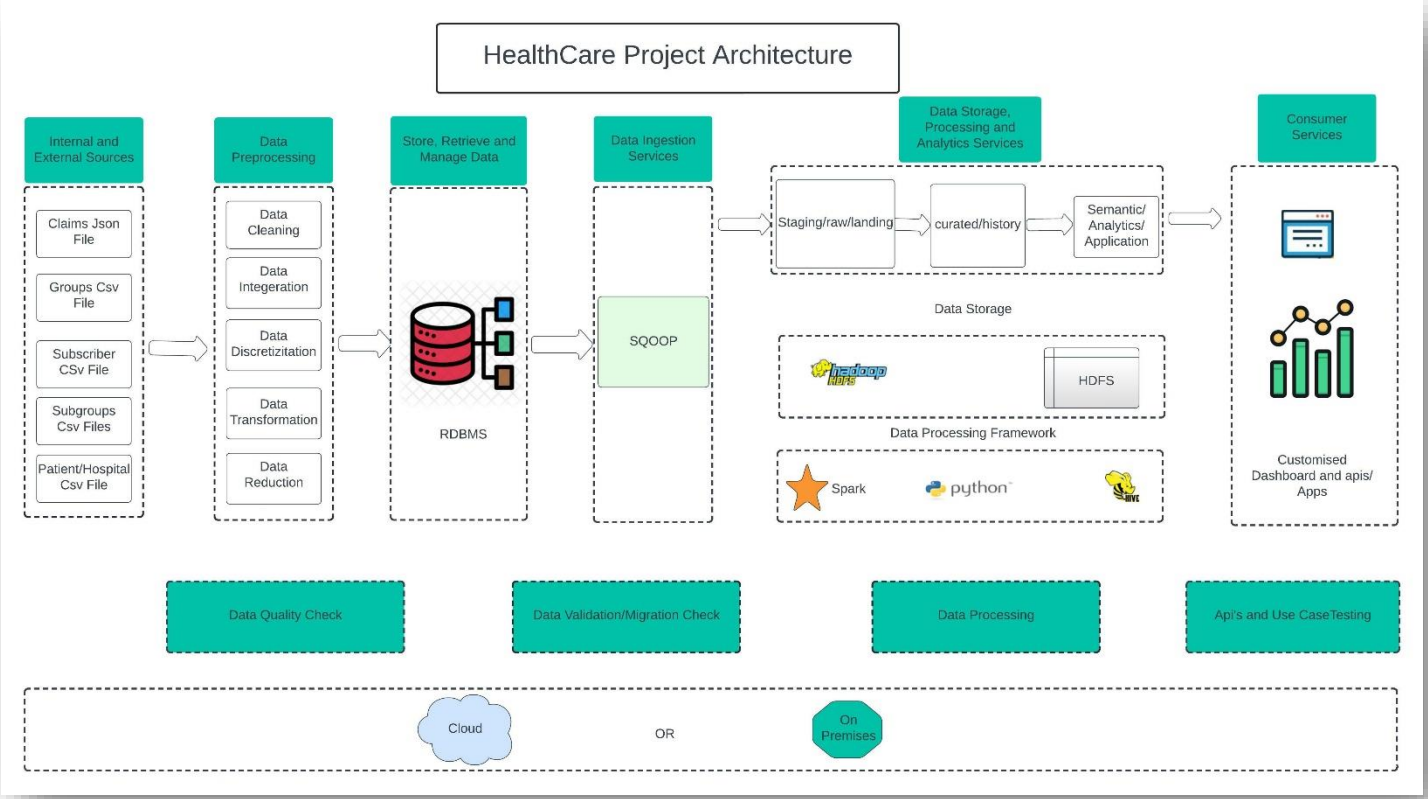
4. After storing the data into the RDBMS, we now transform according to our business requirements.

5. Finally, data landed to HDFS needs to be analyzed by some analytical queries.

6. After analytics queries, we test the result and use the result to enhance the company revenues.

A schematic flow of operations with the best suited components is shown below

3.1 Project Architecture



4. Dataset Explanation & Schema

Data coming from third-party sources reside in local directory and has csv and json format.

Fields present in the data files and tables-

Data files contain the below fields.

Column Name/Field Name Column Description/Field Description

Json File Fields

- CLAIM_ID
- PATIENT_ID
- DISEASE_NAME
- SUB_ID
- CLAIM_DATE
- CLAIM_TYPE
- CLAIM_AMOUNT
- CLAIMED_OR_REJECTED

CSV File 1 Fields (Patient.csv)

- PATIENT_ID
- PATIENT_NAME
- PATIENT _GENDER
- PATIENT _BIRTHDATE
- PATIENT _PHONE
- HOSPITAL_ID
- DISEASE _ NAME
- CITY

CSV File 2 Fields (Subscriber.csv)

- SUB_ID
- FIRST_NAME
- LAST_NAME
- STREET
- BIRTH_DATE
- GENDER
- PHONE_NO
- COUNTRY
- CITY
- ZIP_CODE
- SUBGRP_ID
- ELIG_IND
- E_DATE
- T_DATE

CSV File 3 Fields (Group.csv)

- GRP_ID
- GRP_NAME
- PREMIUM_WRITTEN
- GRP_TYPE
- PIN_CODE
- CITY
- COUNTRY
- ESTABLISHMENT_YEAR

CSV File 4 Fields (disease.csv)

- SUBGRP_ID
- DISEASE_NAME
- DISEASE_ID

CSV File 5 Fields (subgroup.csv)

- SUBGRP_ID
- SUBGRP_NAME
- GRP_ID

CSV File 6 Fields (hospital.csv)

- HOSPITAL_ID
- HOSPITAL _ NAME
- CITY
- STATE
- COUNTRY

CSV File 7 Fields (grpsubgrp.csv)

- SUBGRP_ID
- GRP_ID

5. Problem Statements / Tasks

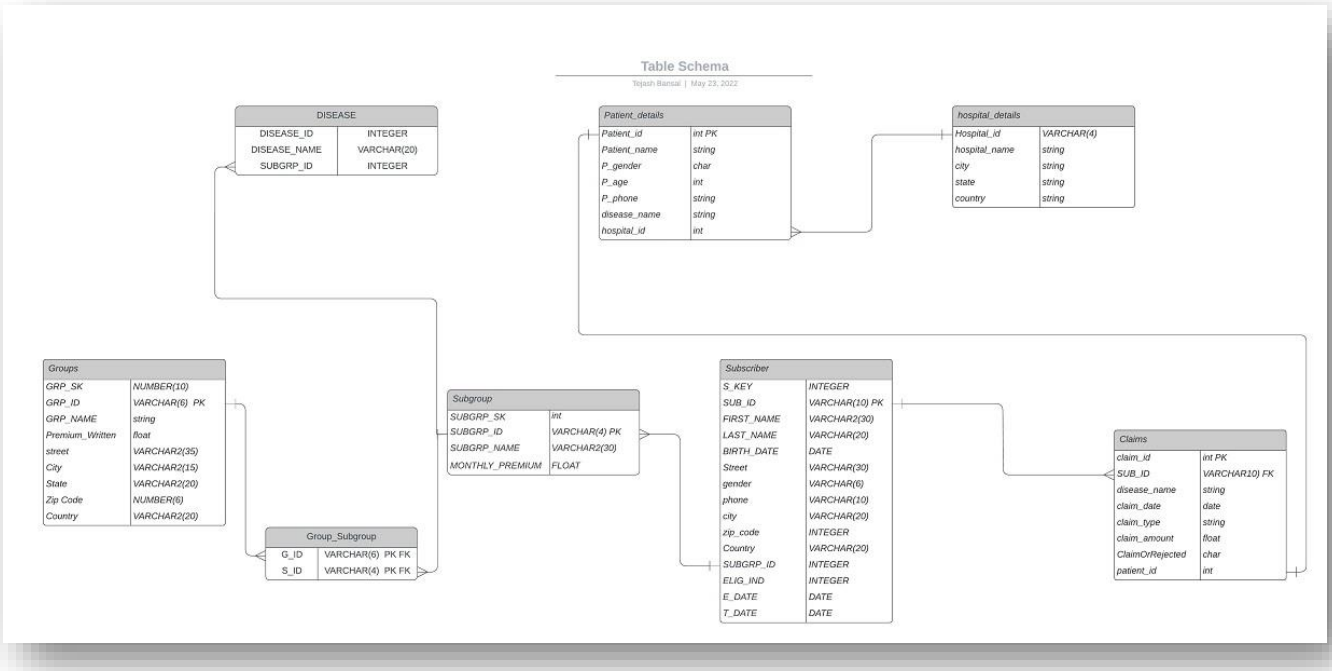
5.1 Problem 1- Data Pre-processing, Enrichment and Load into Database

- Parse and Infer schema of the given xml and csv formats data is ingested.
- We are expected to do general data cleaning steps like empty string replacements with actual NULL, data type checks (including date format) and corrections/ rejections, file name checks, empty file checks, malformed record checks and rejection etc.

Learners must apply below rules for data enrichment process:

- Validate the data from the input file and load only valid records into the target table according to the constraints mentioned in the target table.
- Load only the members who are currently effective. (i.e.) SYSDATE BETWEEN EFFT_DT AND TERM_DT
- Reject records if the Subscriber_Id has less than 9 characters.
- Populate leading zeroes in the fields GROUP_ID and SUBGRP_ID while populating data into the Target table.
- Also validate the Group Id and Subgrp_Id against the Subgrp table and load only matching data into the target table

Schema Design for SQL Database



5.2 Problem 2 - Data Analysis (Spark/Hive)

Once we have made the data ready for analysis, we have to perform below analysis on a batch basis.

1. Find those Subscribers having age less than 30 and they subscribe any subgroup. The output can be in form of a file with columns.

COUNT_OF_SUBSCRIBER

2. Which groups of policies subscriber subscribe mostly Government or private. The output can be in form of a file with columns.

GRP_TYPE,
COUNT(GRP_ID)

3. List female patients over the age of 40 that have undergone knee surgery in the past year. The output can be in form of a file with columns.

PATIENT_NAME

4. Give the Most Profitable subgroup which subscribe the greatest number of times. The output can be in form of a file with columns.

SUBGRP_NAME,
COUNT

5. Give out which groups has maximum subgroups (Policies Groups). The output can be in form of a file with columns.

G_ID,
SUBGRP _COUNT

6. Give the result from where most of the claims are coming (city). The output can be in form of a file with columns.

CITY,
MAX_CLAIM

7. List all the patients whose age is below 18 and who admit for cancer in the hospital. The output can be in form of a file with columns.

PATIENT _ID,
PATIENT _NAME,
AGE

8. List patients who have cashless insura0nce and have total charges greater than or equal for Rs. 50,000. The output can be in form of a file with columns.

PATIENT _ NAME,
PATIENT _GENDER,
PATIENT _BIRTH_DATE

9. Find out total number of claims which were rejected by the groups (insurance companies). The output can be in form of a file with columns.

CLAIM_OR_REJECTED,
COUNT_CLAIM_ID)

10. Give out which disease having maximum number of claims. The output can be in form of a file with columns.

DISEASE_NAME,
COUNT_CLAIMS

Store the above analyzed results as a separate dataset in HDFS.

Approach to Solve

Below steps can be taken to start solving the project problem statements:

- Start by generating Raw Data files in Gateway node location
- Problem 1: Write code to clean & transform data according to the use cases and saved inside the /Processed Data/files folder. After that perform some EDA on top of the cleaned data. Write code and run to take data from /processed data /files and stores all the files in the SQL database using the python and MySQL connector.
- After that we have to write some Sqoop scripts for importing the data from RDBMS System to the HDFS directory /user/hive/warehouse/HEALTHCARE.DB/files
- Write code and run to take data from /user/hive/warehouse/HEALTHCARE.DB/files and solve Problem 2 in a PYSPARK Batch
- Write code and run to take data from '/spark output/files' and perform some visualization on that output files.
- At the end we test all use cases according to the business.

Additional Info

● Submitted code can run in any given Hadoop cluster of the same Hadoop version.

Deliverables

Below are the expected deliverables-
● Code and link to code repository **GitHub Repo**
● Jupiter notebook/ VS code to run the code
● Any other script/wrapper not required to run the code in any environment

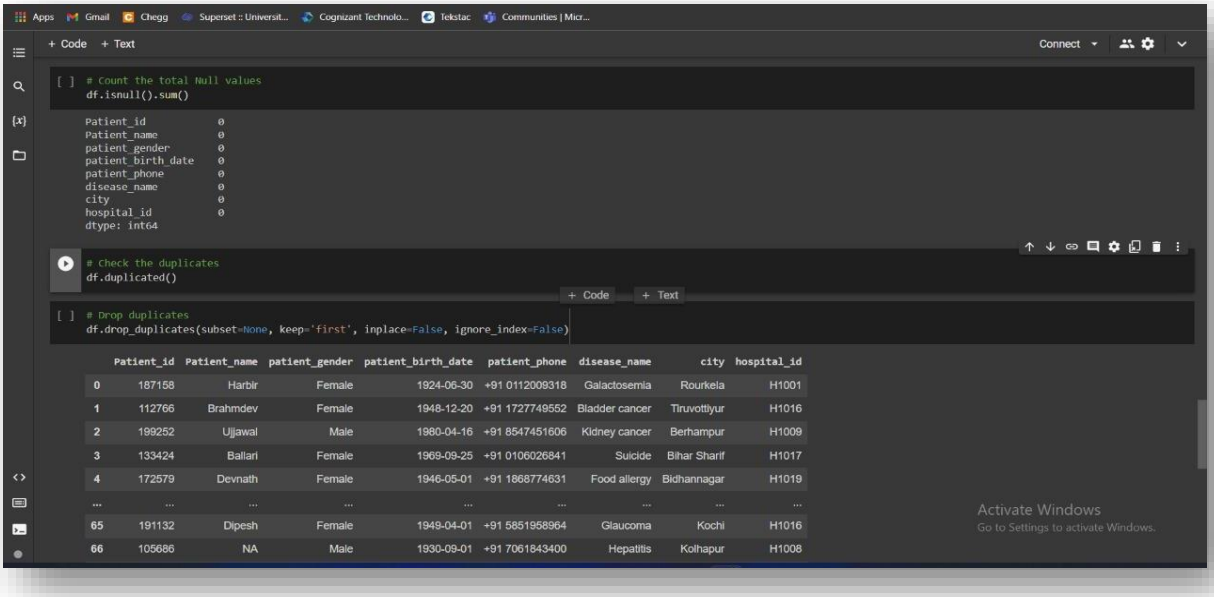
6. Coding/Code Templates:

6.1 Data Processing

6.1.1 Conversion of raw data to processed data:

For each raw file we have checked null values, duplicate values and other parameters and then converted into

processed dataset. here are some samples of codes.



```
# Count the total Null values
df.isnull().sum()
```

```
Patient_id          0
Patient_name        0
patient_gender      0
patient_birth_date  0
patient_phone       0
disease_name        0
city                0
hospital_id         0
dtype: int64
```

```
# Check the duplicates
df.duplicated()
```

```
# Drop duplicates
df.drop_duplicates(subset=None, keep='first', inplace=False, ignore_index=False)
```

	Patient_id	Patient_name	patient_gender	patient_birth_date	patient_phone	disease_name	city	hospital_id
0	187158	Harbir	Female	1924-06-30	+91 0112009318	Galactosemia	Rourkela	H1001
1	112766	Brahmdev	Female	1948-12-20	+91 1727749552	Bladder cancer	Tiruvottiyur	H1016
2	199252	Ujjawal	Male	1980-04-16	+91 8547451606	Kidney cancer	Berhampur	H1009
3	133424	Ballari	Female	1969-09-25	+91 0106026841	Suicide	Bihar Sharif	H1017
4	172579	Devnath	Female	1946-05-01	+91 1868774631	Food allergy	Bidhannagar	H1019
...	...	...	...	...	...	...	...	...
65	191132	Dipesh	Female	1949-04-01	+91 5851958964	Glaucoma	Kochi	H1016
66	105686	NA	Male	1930-09-01	+91 7061843400	Hepatitis	Kolhapur	H1008

Here we are checking for SUB_ID if is it length of 9 or not

```
# Check Subscriber Id not less than 9 digit
count = 0
for i in df['sub _id'].values:

    if len(i)<9:
        df.drop([count], axis=0, inplace=True)
    elif len(str(i))>10:
        df.drop([count], axis=0, inplace=True)
    count = count+1
```

```
# Check the Elig_ind types
df['Elig_ind'].unique()
```

```
array(['Y', 'N'], dtype=object)
```

```
# Check always eff_date is greater than term_date
count = 0
for x,y in df[['eff_date','term_date']].values:
    dob1 = datetime.strptime(x,'%Y-%m-%d').date()
    dob2 = datetime.strptime(y,'%Y-%m-%d').date()
    if dob1 > dob2:
        df.drop([count], axis=0, inplace=True)
    count = count + 1
```

6.1.2 Processed Dataset

Some snippets of processed dataset which is further used to create RDBMS.

patient.csv — Open with

	A	B	C	D	E	F	G	H
1	187158	Harbir	Female	1960-02-24	+91 0112009318	Galactosemia	Rourkela	H1001
2	112766	Brahmdev	Female	1955-05-30	+91 1727749552	Bladder cancer	Tiruvottiyur	H1016
3	199252	Ujjawal	Male	1965-12-31	+91 8547451606	Kidney cancer	Berhampur	H1009
4	133424	Ballari	Female	1979-06-11	+91 0106026841	Suicide	Bihar Sharif	H1017
5	172579	Devnath	Female	1982-02-22	+91 1868774631	Food allergy	Bidhannagar	H1019
6	171320	Atasi	Male	1980-02-21	+91 9747336855	Whiplash	Amravati	H1013
7	107794	Manish	Male	2010-01-26	+91 4354294043	Sunbathing	Panvel	H1004
8	130339	Aakar	Female	1990-04-24	+91 2777633911	Drug consumption	Bihar Sharif	H1000
9	110377	Gurudas	Male	1981-07-16	+91 1232859381	Dengue	Kamarhati	H1001
10	149367	NA	Male	1955-06-03	+91 1780763280	Head banging	Bangalore	H1013
11	156168	NA	Male	2010-02-17	+91 5586075345	Fanconi anaemia	Rajkot	H1004
12	114241	NA	Female	2001-04-30	+91 4146391938	Breast cancer	Ghaziabad	H1015
13	146382	Dharmadaas	Male	1964-10-25	+91 6345482027	Anthrax	Bhalswa Jahangir Pu	H1019
14	132748	Brahmvir	Male	1944-04-04	+91 7316972612	Cystic fibrosis	Ambala	H1018
15	167340	NA	Female	1953-04-04	+91 2960004518	Galactosemia	Surendranagar Dudh	H1003
16	135184	Bhagvan	Female	2011-02-26	+91 0297693485	Dengue	Bhimavaram	H1018
17	179662	Amritkala	Female	1992-04-29	+91 0537157280	Smallpox	Meerut	H1018
18	184479	Bandhu	Male	1981-05-04	+91 0695289163	Pollen allergy	Chinsurah	H1010
19	156988	Bhagavaana	Female	1950-07-31	+91 6071745855	Breast cancer	Shahjahanpur	H1012
20	132870	NA	Female	1959-01-06	+91 8906694405	Glaucoma	Jabalpur	H1017
21	148137	Umang	Female	2017-02-26	+91 9485838770	Pet allergy	Haridwar	H1002
22	113280	Darsana	Male	1951-09-11	+91 7676311811	Rett Syndrome	Dibrugarh	H1019
23	134184	Prakash	Female	1998-06-26	+91 9268324471	Flu	Kottayam	H1001
24	122592	Vaijayanti	Male	1969-04-06	+91 9358851649	Cholera	Mira-Bhayandar	H1009

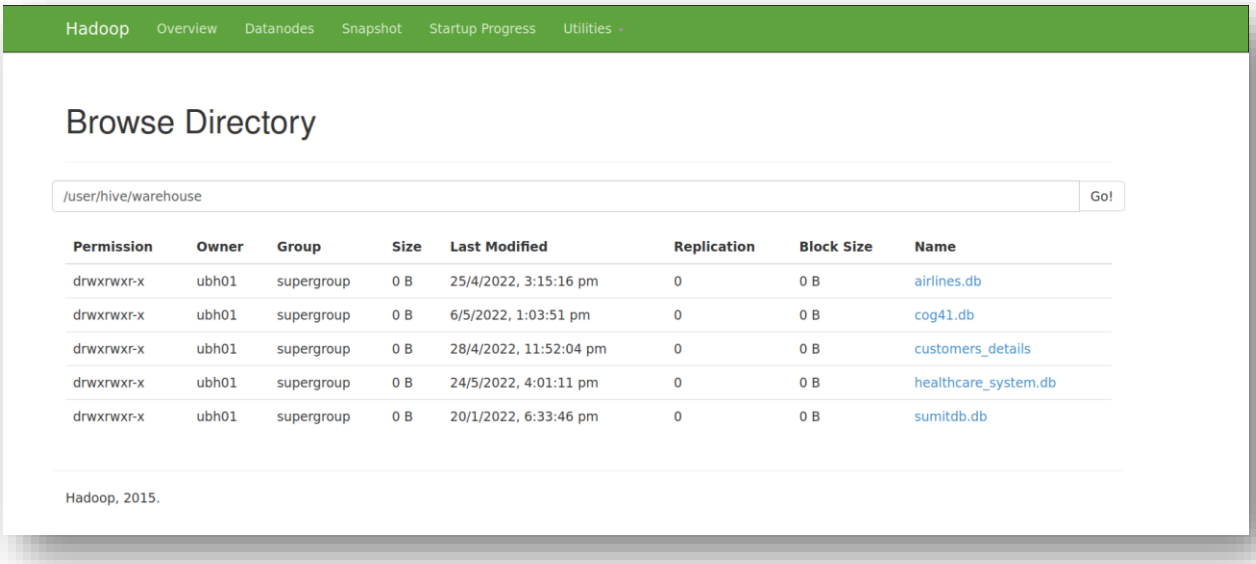
subscriber.csv — Open with

	A	B	C	D	E	F	G	H	I	J	K	L	M
0	SUBID10000	Harbir	Vishwakarma	Baria Marg	1924-06-30	Female	+91 0112009318	India	Rourkela	767058	S107	Y	
1	SUBID10001	Brahmdev	Sonkar	Lala Marg	1948-12-20	Female	+91 1727749552	India	Tiruvottiyur	34639	S105	Y	
2	SUBID10002	Ujjawal	Devi	Mammen Zila	1980-04-16	Male	+91 8547451606	India	Berhampur	914455	S106	N	
3	SUBID10003	Ballari	Mishra	Sahni Zila	1969-09-25	Female	+91 0106026841	India	Bihar Sharif	91481	S104	N	
4	SUBID10004	Devnath	Srivastav	Magar Zila	1946-05-01	Female	+91 1868774631	India	Bidhannagar	531742	S110	N	
5	SUBID10005	Atasi	Seth	Khatri Nagar	1967-10-02	Male	+91 9747336855	India	Amravati	229062	S104	Y	
6	SUBID1006	Manish	Maurya	Swaminathan Chowk	1967-06-06	Male	+91 4354294043	India	Panvel	438733	S109	NA	
7	SUBID10007	Aakar	Yadav	Swamy	1925-03-05	Female	+91 2777633911	India	Bihar Sharif	535907	S104	N	
8	SUBID10008	Gurudas	Gupta	Sarin Nagar	1945-05-06	Male	+91 1232859381	India	Kamarhati	933226	S103	Y	
9	SUBID10009	NA	Gupta	Thakur Circle	1925-06-12	Male	+91 1780763280	India	Bangalore	957469	S105	Y	
10	SUBID1010	NA	Divedi	Dhillon	1976-02-03	Male	+91 5586075345	India	Rajkot	911319	S102	Y	
11	SUBID10011	NA	Vishwakarma	Rajagopalan	1955-01-22	Female	+91 4146391938	India	Ghaziabad	337042	S106	N	
12	SUBID10012	Dharmadaas	Tiwari	Rama	1964-04-29	Male	+91 6345482027	India	Bhalswa Jahangir Pu	430793	S103	N	
13	SUBID10013	Brahmvir	Rai	Shah Path	1991-11-11	Male	+91 7316972612	India	Ambala	249898	S106	N	
14	SUBID10014	NA	Srivastav	Chandra Path	1981-01-25	Female	+91 2960004518	India	Surendranagar Dudh	111968	S102	N	
15	SUBID10015	Bhagvan	Srivastav	Edwin	1966-07-24	Female	+91 0297693485	India	Bhimavaram	436513	S105	Y	
16	SUBID10016	Amritkala	Srivastav	Guha Path	1933-11-20	Female	+91 0537157280	India	Meerut	863467	S106	Y	
17	SUBID10017	Bandhu	Seth	Varughese	1996-10-15	Male	+91 0695289163	India	Chinsurah	136713	S108	N	
18	SUBID10018	Bhagavaana	Kumar	Kulkarni Zila	1935-09-16	Female	+91 6071745855	India	Shahjahanpur	597276	S101	N	
19	SUBID10019	NA	Maurya	Sharaf Nagar	1924-11-09	Female	+91 8906694405	India	Jabalpur	958538	S104	N	
20	SUBID1020	Umang	Srivastav	Balay Chowk	1963-07-14	Female	+91 9485838770	India	Haridwar	181692	S109	Y	
21	SUBID1021	Darsana	Yadav	Upadhyay Zila	1932-05-29	Male	+91 7676311811	India	Dibrugarh	187414	S109	Y	
22	SUBID10022	Prakash	Rao	Sachar	1923-09-15	Female	+91 9268324471	India	Kottayam	180680	S104	N	
23	SUBID10023	Vaijayanti	Pratap	Khalsa Nagar	1920-11-13	Male	+91 9358851649	India	Mira-Bhayandar	419190	S102	Y	

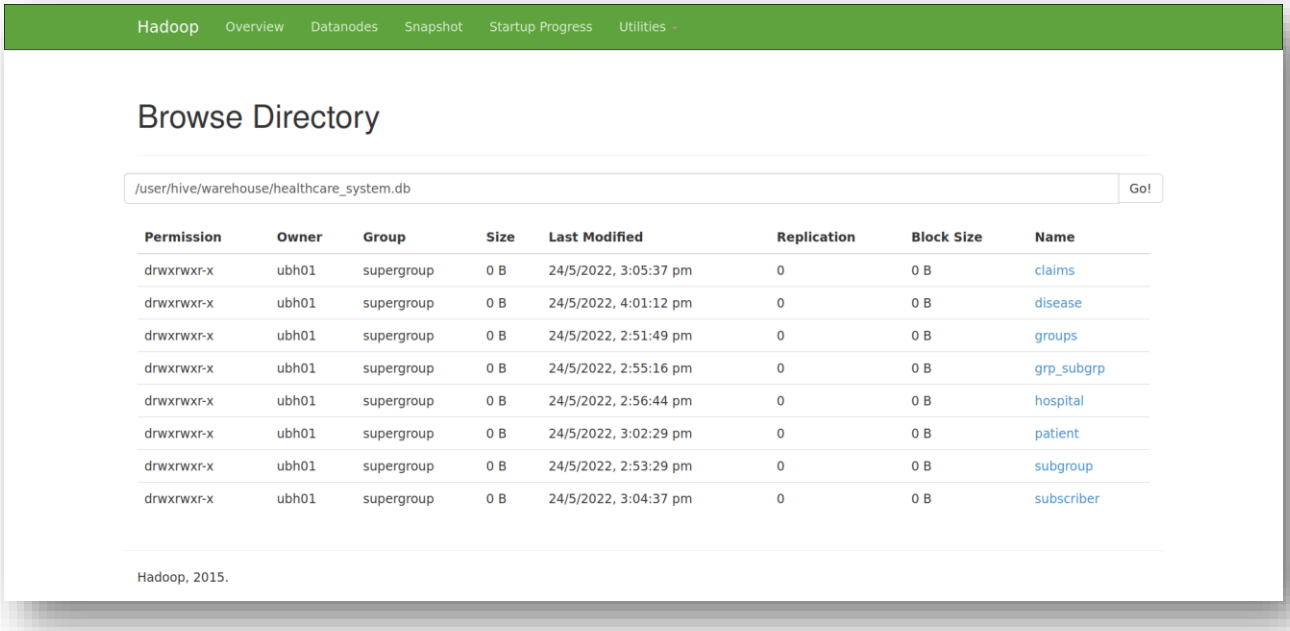
6.2 Hive and Sqoop

We have used Sqoop to import the data form RDBMS to Hive and there we can perform our necessary tasks to get the outputs

Here is the HEALTHCARE_SYSTEM Database created in Hive.



The tables created in the databases as mentioned in the schema



6.3 Apache Spark

After uploading the data in to HDFS we connected spark. Here we analyze the data with help of python. Here we get our desired result in tabular form and that result is used to visualize our use cases.

Some snippet of the following code and result-

```
# List patients who have cashless insura0nce and have total charges greater than or equal for Rs. 50,000.

sparkdf = spark.sql("select patient_name,patient_gender,patient_birth_date \
          from patient join claims on patient.patient_id = claims.patient_id \
          where claim_amount >= 50000 and claim_type = 'claims of value'")
sparkdf.toPandas().to_csv('Spark Outputs for Visualization/query13.csv')
sparkdf.show()
```

```
+------------+--------------+------------------+
|patient_name|patient_gender|patient_birth_date|
+------------+--------------+------------------+
|   Anjushree|          Male|        1982-06-28|
|  Chitranjan|        Female|        2020-10-27|
|      Gensho|          Male|        1991-07-27|
|  Vaijayanti|          Male|        1969-04-06|
|       Aakar|        Female|        1990-04-24|
|          NA|        Female|        1959-01-06|
|       Saroj|        Female|        1953-07-21|
|     Bhagvan|        Female|        2011-02-26|
|  Dharmadaas|          Male|        1964-10-25|
|       Umang|        Female|        2017-02-26|
|          NA|          Male|        1955-06-03|
|      Kishan|          Male|        1955-06-30|
|          NA|        Female|        1953-04-04|
|     Devnath|        Female|        1982-02-22|
|      Harbir|        Female|        1960-02-24|
|       Ekant|          Male|        1969-11-01|
|          NA|          Male|        2013-10-30|
|          NA|          Male|        1956-04-04|
|       Lalit|        Female|        1978-04-30|
|     Ujjawal|          Male|        1965-12-31|
+------------+--------------+------------------+
```

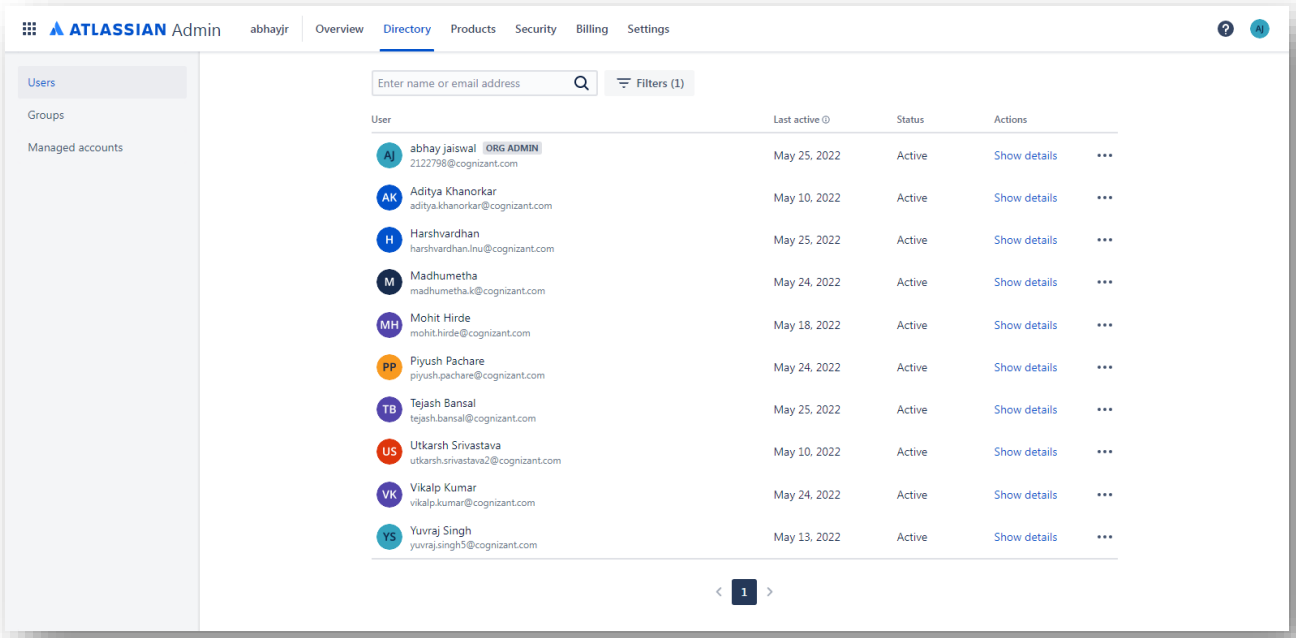
```
+ Code + Text  Last saved at May 24                                   Connect

[ ] # Find out hospital which serve most number of patients
    sparkdf = spark.sql("select hospital_name,count(patient_id) as total_patient \
              from hospital join patient on hospital.hospital_id=patient.hospital_id \
              group by hospital_name order by total_patient desc")

    sparkdf.toPandas().to_csv('Spark Outputs for Visualization/query4.csv')
    sparkdf.show()
```

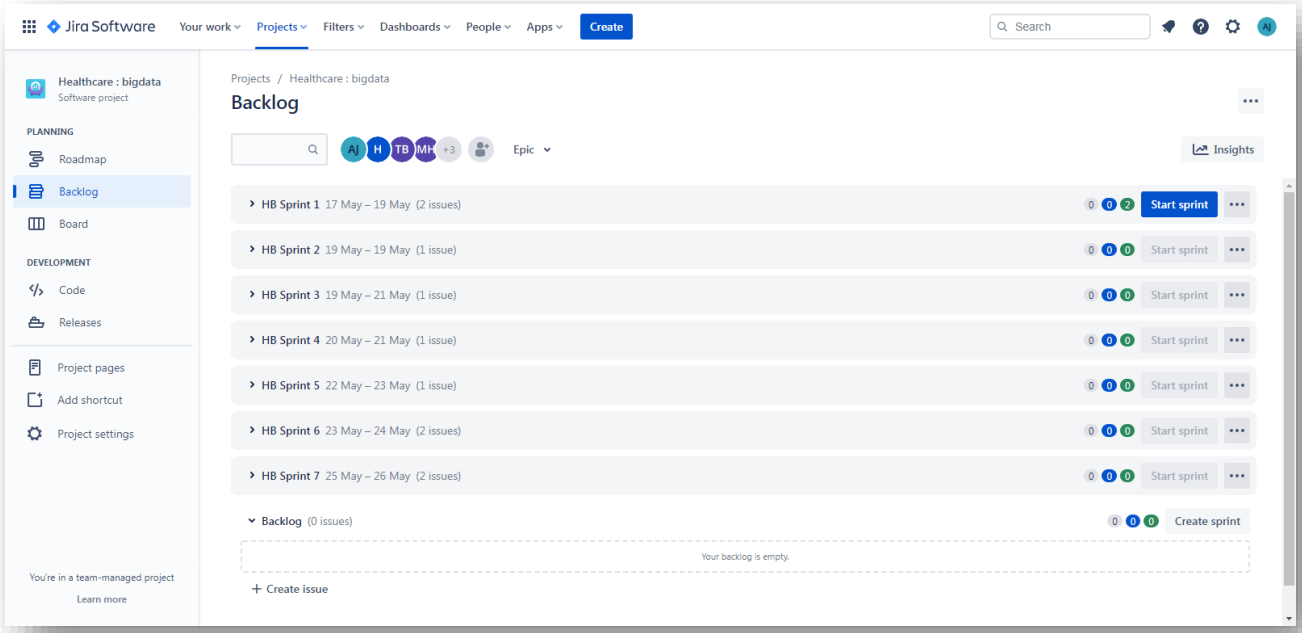
```
+--------------------+-------------+
|       hospital_name|total_patient|
+--------------------+-------------+
|   Manipal Hospitals|            9|
|Apollo Hospitals ...|            8|
|Medanta The Medicity|            7|
|Jaslok Hospital a...|            6|
|Indraprastha Apol...|            5|
|PGIMER - Postgrad...|            4|
|Apollo Hospital -...|            4|
|Fortis Hospital M...|            4|
|King Edward Memor...|            3|
|Apollo Health Cit...|            3|
|Yashoda Hospital ...|            3|
|Bombay Hospital &...|            3|
|Fortis Hiranandan...|            2|
|Lilavati Hospital...|            2|
|The Christian Med...|            2|
|Fortis Flt. Lt. R...|            1|
|P. D. Hinduja Nat...|            1|
|Breach Candy Hosp...|            1|
|All India Institu...|            1|
|Sir Ganga Ram Hos...|            1|
+--------------------+-------------+
```

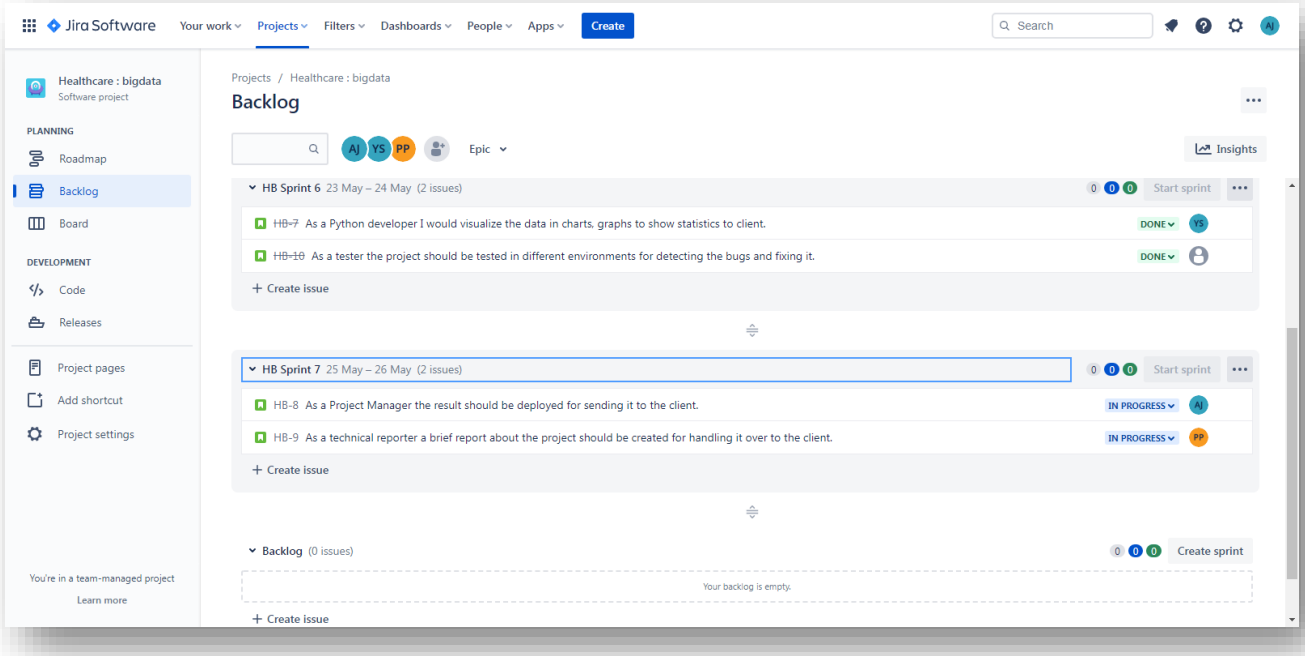
7. Project Management Tool

Jira Software is part of a family of products designed to help teams of all types manage work. Originally, Jira was designed as a bug and issue tracker. But today, Jira has evolved into a powerful work management tool for all kinds of use cases, from requirements and test case management to agile software development.

In this project we use Jira as a Project Management tool. With the help of the Jira, we assign a task and customizes the issues and subtasks in a whole team easily, also manages the workflow and track the progress. It also helps us to change the permission for a particular task within the team.



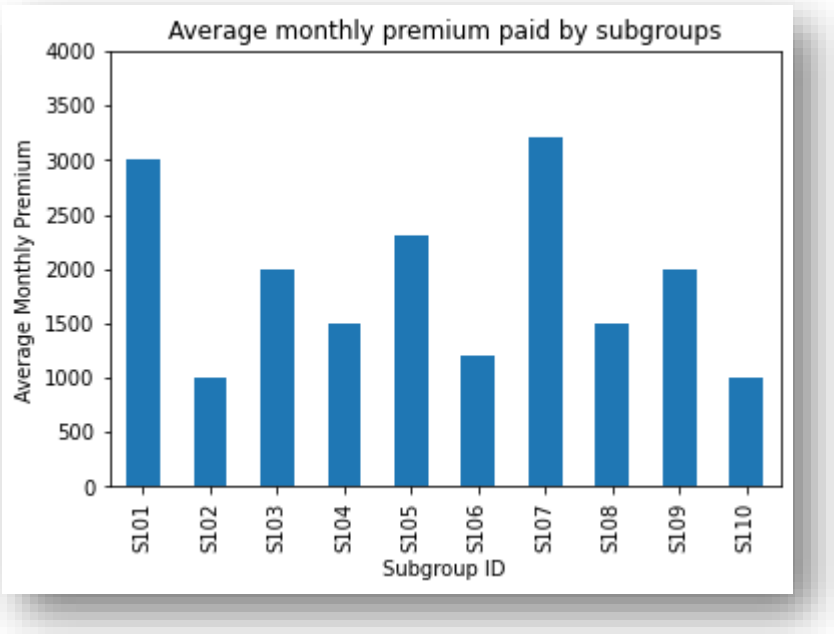

Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Healthcare : bigdata
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Releases
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Projects / Healthcare : bigdata
Backlog
Epic
Insights
HB Sprint 1 17 May – 19 May (2 issues)
Start sprint
HB Sprint 2 19 May – 19 May (1 issue)
HB Sprint 3 19 May – 21 May (1 issue)
HB Sprint 4 20 May – 21 May (1 issue)
HB Sprint 5 22 May – 23 May (1 issue)
HB Sprint 6 23 May – 24 May (2 issues)
HB Sprint 7 25 May – 26 May (2 issues)
Backlog (0 issues)
Create sprint
Your backlog is empty.
Create issue


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Healthcare : bigdata
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Releases
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Projects / Healthcare : bigdata
Backlog
Epic
Insights
HB Sprint 6 23 May – 24 May (2 issues)
Start sprint
HB-7 As a Python developer I would visualize the data in charts, graphs to show statistics to client.
DONE
HB-10 As a tester the project should be tested in different environments for detecting the bugs and fixing it.
DONE
Create issue
HB Sprint 7 25 May – 26 May (2 issues)
HB-8 As a Project Manager the result should be deployed for sending it to the client.
IN PROGRESS
HB-9 As a technical reporter a brief report about the project should be created for handling it over to the client.
IN PROGRESS
Backlog (0 issues)
Create sprint
Your backlog is empty.
Create issue

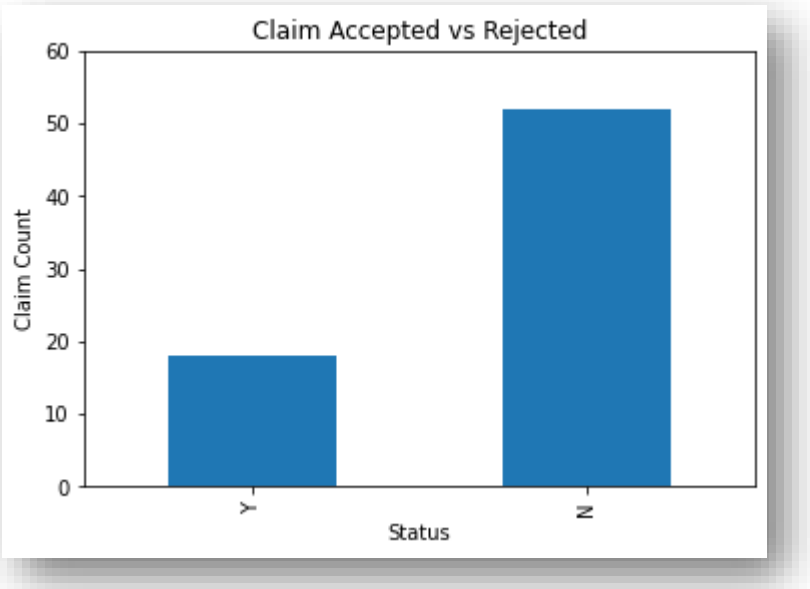
8. Output Screens

We used Matplotlib and seaborn to visualize our use cases which will be better to take business decision.

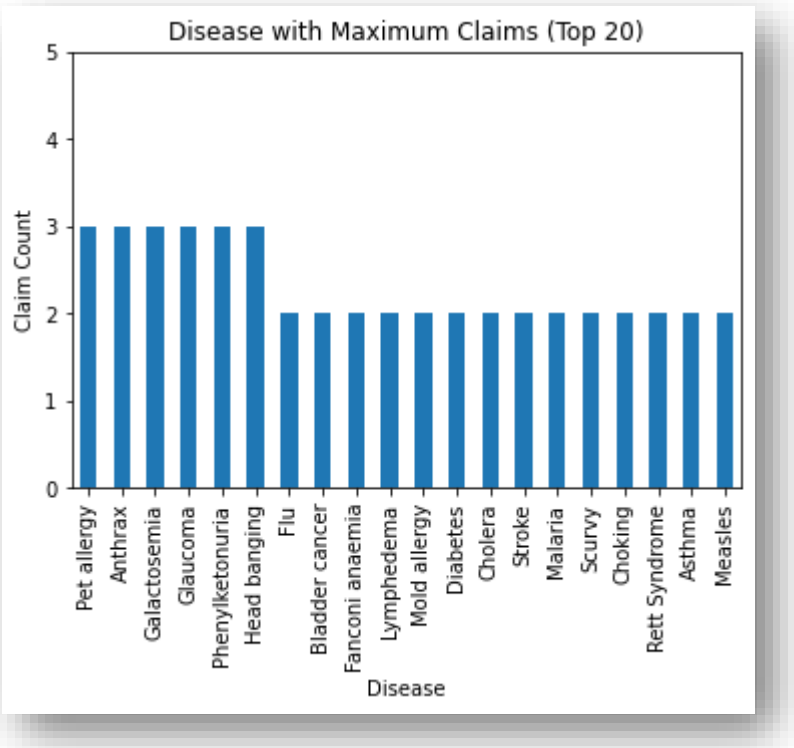
Use Case-1: Average Monthly premium for each subgroup



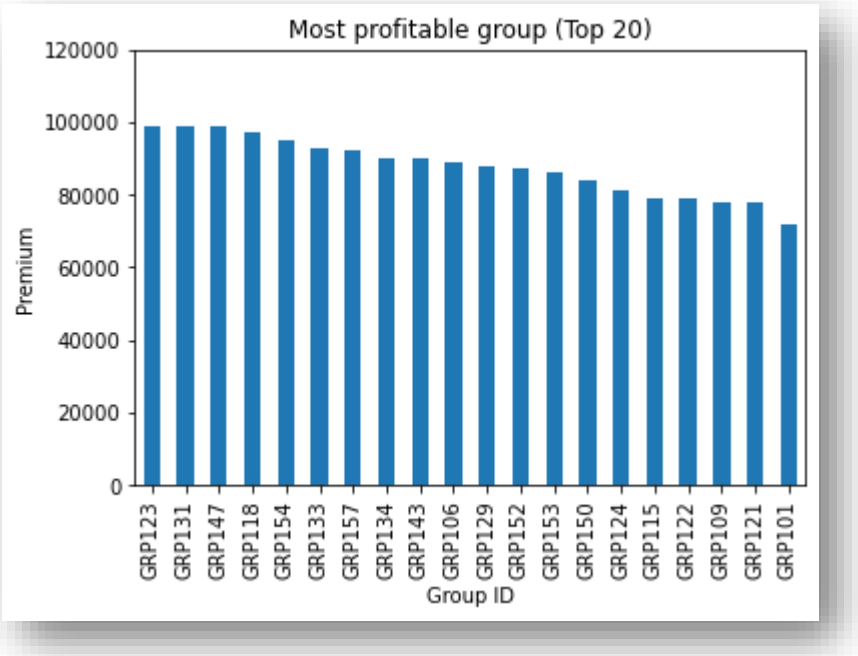
Use Case-2: Number of people whose claim either got accepted or rejected.



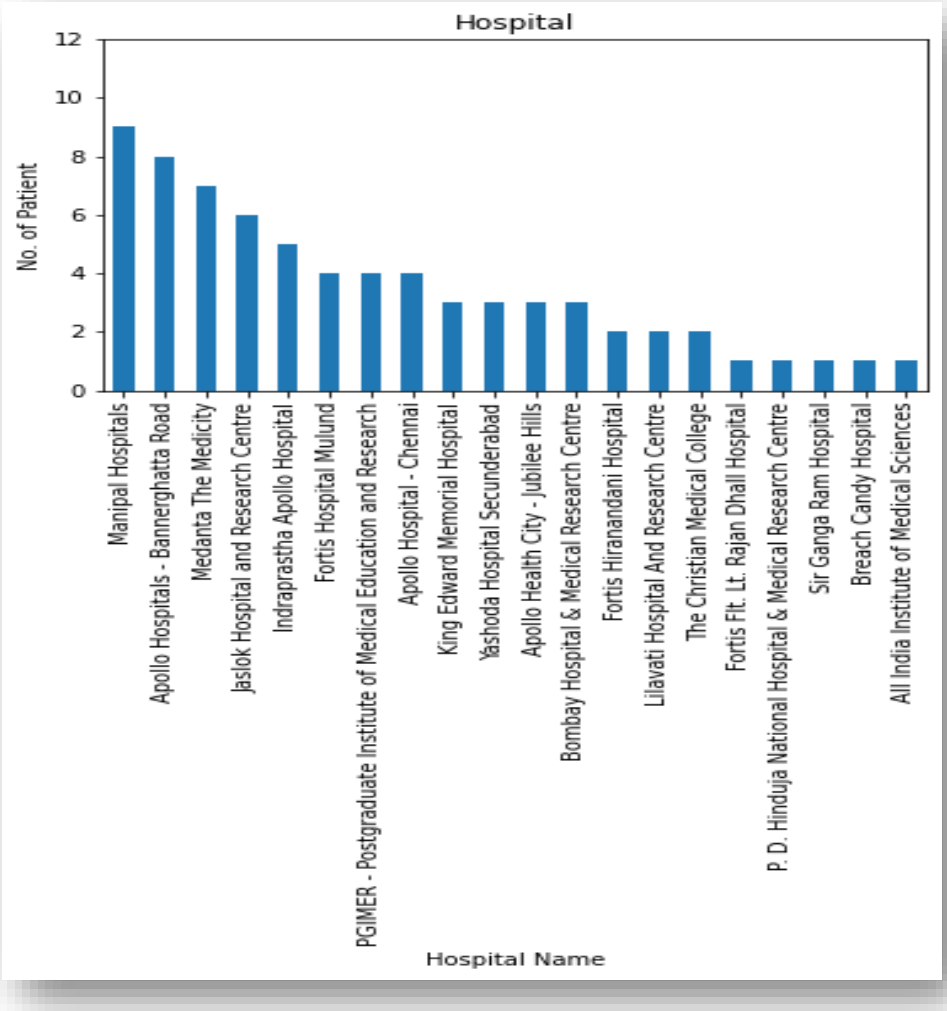
Use case-3: Which disease have maximum number of claims



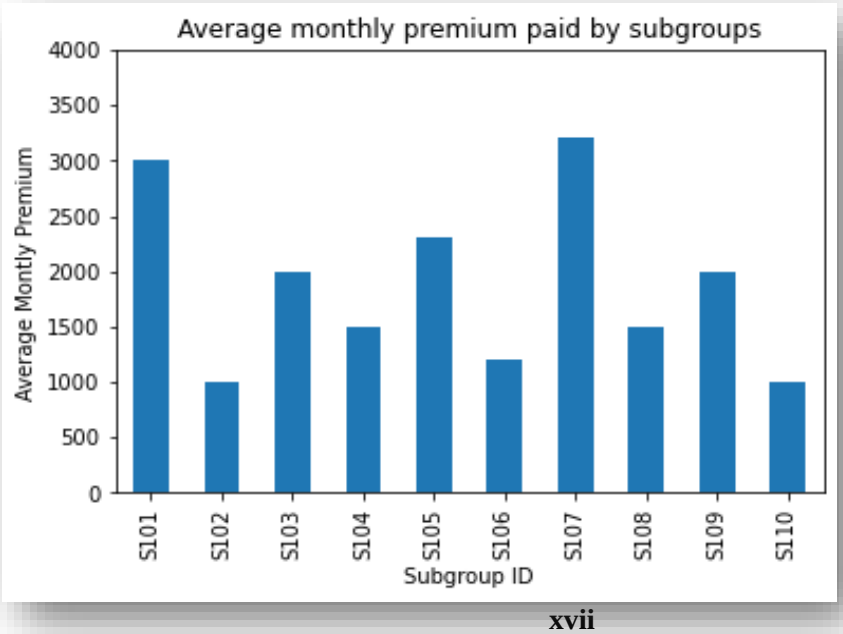
Use Case-4: Which company/group is most profitable



Use case-5: No. of patient in each hospital



Use case-6: Average Monthly premium paid by each subgroup.



9. Conclusion

We have collected data from various 3rd party sources and processed them and with the help of Big Data tools we computed the data to visualize some of necessary use case. Based on the above analysis the health care insurance company will create a new business strategy to acquire more customers, engagement and send offers. As well as fetching the company and customer details and provide easy access to information regarding customers.

10. Further Enhancements/Recommendations

This project has a very vast scope in future in this field. We developed this project on the requirement of our client but it can be generalized in future. If we get required resources, we can get more accurate results. There are various use cases that can be achieved by this project. Some of future scopes are bellow-

- Real time data can also be used for real time processing.
- We can automate the whole procedure where data coming from sources and getting executed at a same time.
- Not in the Healthcare industry we can generalized the whole procedure to other sectors like cars, online education system etc.

11. References/Bibliography

Books and Reports
[1] Beranger, Jérôme. 2016. *Ethics in Big Data: the medical datasphere.* **London: Elsevier.**
[2] Davis, Cord and Patterson, Doug. 2012. Ethics of Big Data. Farnham, O'Reilly.

Web Sites
[1] Big Data Ethics
[2] GeekforGeeks