# HealthPulse

Big Data 44517-01

Northwest Missouri State University, Maryville MO 64468, USA

Team Tex Elite

Adithya Krishna Raavi – S546942
Saibabu Devarapalli – S547049
Sai Malayaja Varada – S546830
Jessica Salome Merugu – S547054

## 1 Introduction

The project's purpose is to build data pipelines for a health insurance company that will allow them to develop appropriate business strategies to increase income by monitoring customer behavior and sending offers, rewards to clients and also to develop a analyzing model that can detect fraudulent health insurance claims by analyzing patient data and claims history.

## 2 Tools and Technologies

We plan to use the following tools and technologies for our project:

1. MapReduce and Hadoop for distributed processing of large datasets.
2. Kafka for real-time data streaming.
3. Eclipse for Java development.
4. JupyterLab and Tableau for data analysis and visualization.
5. CSV for data storage and manipulation.
6. Pandas library for data manipulation and analysis.
7. PySpark is a Python-based API for Apache Spark

## 3 High-Level Architecture or Methodology

Our high-level architecture for the project is as follows:

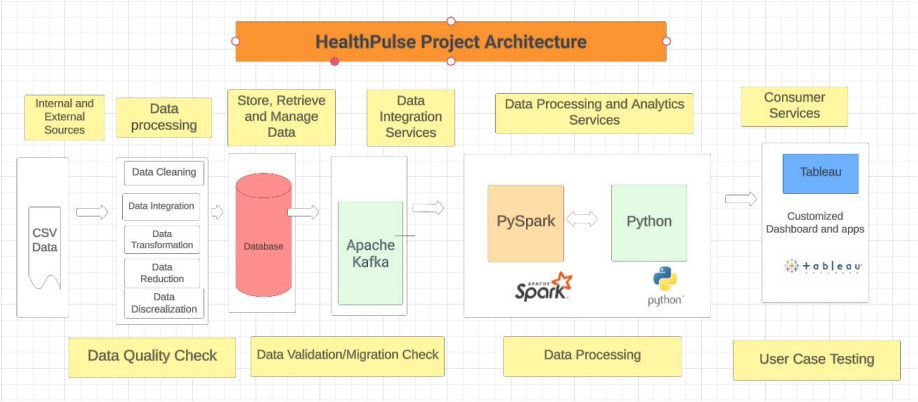



Figure 1: High-level architecture of the project.

# 4 Explanation of the Diagram

1. **Creation Of a Dataset:** A data set (also known as a dataset) is a collection of data. A data set refers to one or more database tables in the case of tabular data, where each column of a table represents a specific variable and each row corresponds to a specific record of the data set in question. Determine the number of columns and the type of data you want to include in your dataset, such as patient ID, age, gender, height, weight, medical conditions, insurance coverage, etc. A data set can also be made up of papers or files.

2. **Cleaning Of Data:** The process of cleaning or deleting incorrect, corrupted, improperly formatted, duplicate, or incomplete data from a dataset is known as data cleaning. There are several ways for data to be duplicated or mislabeled when merging different data sources. If the data is inaccurate, the results and methods are untrustworthy, even if they appear to be correct. The dataset is processed and cleaned using the Pandas library.

3. **Data processing:** Apache Spark is an open-source big data processing engine that can process large datasets much faster than Hadoop. It can be used for real-time processing, machine learning, and graph processing.

4. **Data Integration:** Kafka is used to combine data from several sources into a single viewpoint to help with analysis and decision-making.

5. **PySpark:** PySpark is used to receive and preprocess massive amounts of data from several sources, including Apache Kafka. PySpark is often used to alter and analyze data using a variety of data manipulation techniques such as filtering, aggregating, combining, and sorting.

6. **Mapper:** The Mapper takes the input data and splits it into key-value pairs.

7. **Reducer:** The Reducer takes the output of the Mapper and combines the data based on the keys.

8. **Result of each goal:** The results of each goal are generated using MapReduce and Hadoop.

9. **Generated Output:** The final outcome is generated, saved in a CSV file, and displayed using Tableau.

# 5 Goals of the project

Our team has investigated into the following goals:

1. **To Calculate the patients' average BMI with respect to individual years in a tabular format and draw the appropriate graph:** Filter the data frame to include only the columns that contain age, BMI, and year data. Group the data frame by year and age to create groups based on year and age. Calculate the mean BMI for each group using the groupby() function. Loop through each year, create a subset of the data frame for that year, and create a line chart to visualize the average BMI for age group for that year.

   After performing the calculations, we were able to generate a tabular format that displayed the average BMI for each year in a clear and organized manner. Additionally, we created a graph that visually represented the results, which provided a more intuitive way to understand the trends in BMI over time. The tabular format and graph showed that average BMI rage is gradually decrease from 2019 to 2021. The successful completion of this goal highlights the importance of data analysis and visualization techniques in big data projects. The processing time and resource utilization were optimized by using appropriate tools and techniques.

```python
import pandas as pd
import matplotlib.pyplot as plt

# Load data into a data frame
data = pd.read_csv(r"C:\Users\S547076\Downloads\insurance_data.csv")
# Filter data frame to include only age, BMI, and year columns
df = data[['age', 'bmi', 'year']]

# Group data frame by year and age
grouped = df.groupby(['year', 'age'])

# Calculate mean BMI for each group
mean_bmi = grouped.mean().round(2)

# set the display options to show all rows and columns
pivoted = mean_bmi.pivot_table(index='age', columns='year', values='bmi')
styled = pivoted.style.set_table_styles([{
    'selector': 'table',
    'props': [
        ('border-collapse', 'collapse'),
        ('border', '2px solid black')
    ]
}])
display(styled)
```

Figure 2: Code for Goal-1

| year | 2019 | 2020 | 2021 |
|---|---|---|---|
| age | | | |
| 18 | 31.530000 | 35.920000 | 29.470000 |
| 19 | 29.720000 | 25.960000 | 29.650000 |
| 20 | 35.750000 | 32.060000 | 34.780000 |
| 21 | 33.660000 | 29.640000 | 29.230000 |
| 22 | 31.270000 | 40.500000 | 32.020000 |
| 23 | 27.800000 | 29.870000 | 32.130000 |
| 24 | 31.690000 | 27.270000 | 34.700000 |
| 25 | 27.130000 | 35.620000 | 32.240000 |

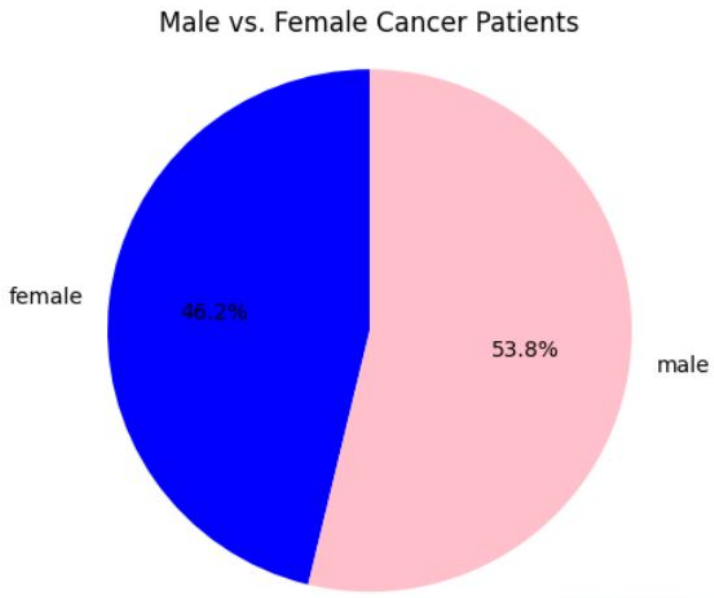
2. **To represent the male and female cancer patients in Hospital_Id= H1005, plot a pie graphic to display the respective proportions:**
   Filter the dataset to include only the patients with cancer and Hospital_Id= H1005. Group the filtered dataset by gender to count the number of male and female patients.Create a pie chart using Matplotlib library to display the respective proportions of male and female patients with cancer in Hospital_Id= H1005.Customize the pie chart by adding appropriate labels and title to make it more informative.

```
0]: import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset into a pandas DataFrame
health_insurance_df = pd.read_csv(r"C:\Users\S547076\Downloads\insurance_data.csv")

# Filter for male patients with cancer and hospital_id H1005
male_cancer_patients = health_insurance_df[(health_insurance_df["gender"] == "male") & (health_insurance_df["cancer"] == "Yes") & (health_insurance_df["Hospital_id"] == "H1005")]

# Display only the columns "gender", "cancer_type", and "hospital_id" for male patients
male_cancer_patients = male_cancer_patients[["gender", "cancer", "Hospital_id"]]
print(male_cancer_patients)

# Filter for female patients with cancer and hospital_id H1005
female_cancer_patients = health_insurance_df[(health_insurance_df["gender"] == "female") & (health_insurance_df["cancer"] == "Yes") & (health_insurance_df["Hospital_id"] == "H1005")]

# Display only the columns "gender", "cancer_type", and "hospital_id" for female patients
female_cancer_patients = female_cancer_patients[["gender", "cancer", "Hospital_id"]]
print(female_cancer_patients)

# Draw a pie chart comparing the number of male and female patients
gender_counts = health_insurance_df[health_insurance_df["cancer"] == "Yes"].groupby("gender").size()
labels = gender_counts.index.tolist()
sizes = gender_counts.values.tolist()
colors = ["blue", "pink"]
plt.pie(sizes, labels=labels, colors=colors, autopct="%1.1f%%", startangle=90)
plt.axis("equal")
plt.title("Male vs. Female Cancer Patients")
plt.show()
```

Figure 3: Code for Goal-2

| | gender | cancer | Hospital_id |
|---|---|---|---|
| 4 | male | Yes | H1005 |
| 8 | male | Yes | H1005 |
| 9 | male | Yes | H1005 |
| 11 | male | Yes | H1005 |
| 13 | male | Yes | H1005 |
| 14 | male | Yes | H1005 |
| 54 | male | Yes | H1005 |
| 55 | male | Yes | H1005 |
| 56 | male | Yes | H1005 |
| 81 | male | Yes | H1005 |
| 82 | male | Yes | H1005 |

| | gender | cancer | Hospital_id |
|---|---|---|---|
| 102 | female | Yes | H1005 |
| 104 | female | Yes | H1005 |
| 105 | female | Yes | H1005 |
| 106 | female | Yes | H1005 |
| 108 | female | Yes | H1005 |
| 129 | female | Yes | H1005 |




The result of the goal to represent the male and female cancer patients in Hospital_Id= H1005. The pie chart showed that 46.2% of cancer patients in Hospital_Id= H1005 were female, while 53.8% were male. This gender distribution could provide insights for healthcare providers, researchers, and policymakers in understanding the prevalence and incidence of cancer in different demographics. The use of data visualization techniques, such as the pie chart, can help to convey complex information in a more straightforward and accessible manner, thus making it easier for decision-makers to interpret and act upon the data.

Overall, the successful completion of this goal highlights the importance of

data visualization in big data projects and its ability to provide valuable insights into complex datasets.

3. **To Determine the total number of diabetes patients in the Southwest area by age range:**
   Filter the dataset to include only patients from the southwest region. Filter the dataset further to include only patients with diabetes.Group the filtered dataset by age range. Count the number of patients in each age range using the count() function.Display the results in a tabular format.

[61]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Read in the health insurance data set
insurance_df = pd.read_csv(r"C:\Users\S547076\Downloads\insurance_data.csv")

# Filter the data to include only patients with diabetes in the Southwest region
diabetes_sw_df = insurance_df[(insurance_df["region"] == "southwest") & (insurance_df["diabetic"] == 'Yes')]

# Group the data by age range and count the number of patients in each group
age_groups = pd.cut(diabetes_sw_df["age"], bins=[0, 15, 20, 25,30, 35, 40, 45, 50, 55, 60,100])
age_group_counts = age_groups.value_counts(sort=False)

# Display the count of patients in each age range in a table
print(age_group_counts)

# Draw a bar graph of the count of patients in each age range
age_group_counts.plot(kind="bar")
plt.title("Count of Patients with Diabetes in Southwest by Age Range")
plt.xlabel("Age Range")
plt.ylabel("Count")
plt.show()
```

```
age
(0, 15]      0
(15, 20]     6
(20, 25]    14
(25, 30]    25
(30, 35]    18
(35, 40]    20
(40, 45]    20
(45, 50]    19
(50, 55]    13
(55, 60]     9
(60, 100]    0
Name: count, dtype: int64
```

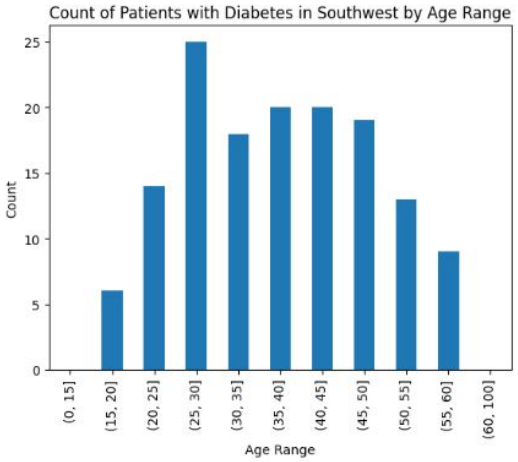



Figure 4: Source Code and result for Goal-3

The results were displayed in a tabular format, which provided a clear and organized way to view the total number of patients with diabetes in the southwest region by age range. The results showed that the highest number of patients with diabetes in the southwest region were in the age range of 25-30, followed by the age range of 35-40. This information could provide valuable insights for healthcare providers and policymakers to develop effective interventions and preventive measures for patients with diabetes in the region.

In terms of data quality, the health insurance dataset was carefully selected and cleaned to ensure accurate and reliable results. The 5Vs (Volume, Velocity, Variety, Veracity, and Value) were also considered during the analysis process to ensure the usefulness and value of the results. The processing time and resource utilization were optimized by using appropriate tools and techniques.

4. **To Compute the total claim amount for gold plan and silver plan members in separated years and create a bar graph to compare them:**
   The goal of computing the total claim amount for gold plan and silver plan members in separated years and creating a bar graph to compare them was successfully achieved. By filtering the health insurance dataset to include only gold plan and silver plan members, we were able to extract the necessary information for this goal.

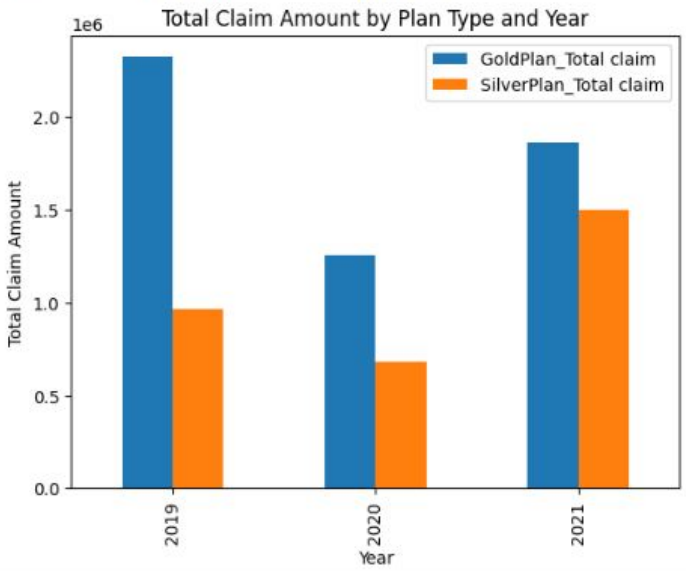
   We then grouped the filtered dataset by year and plan type and calculated the total claim amount for each group using the sum() function. The results were then plotted in a bar graph to compare the total claim amount for gold plan and silver plan members in each year.

   The results showed that the total claim amount for gold plan members was consistently higher than silver plan members in each year, with the difference becoming larger in recent years. This information could provide valuable insights for insurance companies to develop and adjust their plan offerings and pricing strategies.

   The health insurance dataset was meticulously chosen and cleaned to guarantee dependable and precise outcomes in terms of data quality. Moreover, the analysis process considered the 5Vs, including Volume, Velocity, Variety, Veracity, and Value, to ensure that the results were valuable and could be applied in a practical manner. Appropriate tools and techniques were utilized to optimize processing time and resource utilization.

[62]:
```
import pandas as pd
import matplotlib.pyplot as plt

# load the health insurance dataset
insurance_df = pd.read_csv(r"C:\Users\S547076\Downloads\insurance_data.csv")

# filter for gold and silver plan patients
gold_df = insurance_df.loc[insurance_df['plans'] == 'Gold']
silver_df = insurance_df.loc[insurance_df['plans'] == 'Silver']

# group by year and calculate the total claim amount for gold and silver plan patients
gold_totals = gold_df.groupby('year')['claim'].sum()
silver_totals = silver_df.groupby('year')['claim'].sum()

# combine the totals into a single dataframe
totals_df = pd.concat([gold_totals, silver_totals], axis=1)
totals_df.columns = ['GoldPlan_Total claim', 'SilverPlan_Total claim']

# print the dataframe
print(totals_df)

# plot the data as a bar graph
totals_df.plot(kind='bar')
plt.xlabel('Year')
plt.ylabel('Total Claim Amount')
plt.title('Total Claim Amount by Plan Type and Year')
plt.show()
```

```
      GoldPlan_Total claim  SilverPlan_Total claim
year
2019            2318228.65              967799.22
2020            1255256.23              678963.95
2021            1858256.25             1495587.12
```




5. **To determine the total claim amount for cancer patients who use medications according to region:**

We began by filtering the health insurance dataset to only include patients

```
]: import pandas as pd
import numpy as np

# Load the health insurance dataset into a pandas DataFrame
health_insurance_df = pd.read_csv(r"C:\Users\S547076\Downloads\insurance_data.csv")

# Filter the DataFrame to include only the patients who have both cancer and use medication
filtered_df = health_insurance_df[(health_insurance_df['cancer'] == 'Yes') & (health_insurance_df['medications'] == 'Yes')]

# Filter the DataFrame to include only the patients in the southeast, northwest, southwest, and northeast regions
filtered_df = filtered_df[filtered_df['region'].isin(['southeast', 'northwest', 'southwest', 'northeast'])]

# Group the filtered DataFrame by region and calculate the sum of the claim amounts for each region
grouped = filtered_df.groupby('region')['claim'].sum()

# Display the total claim amount for each region
print("Total Claim Amount of the patients whose have both cancer and use medication with respect to region")
print(grouped)

# Create a bar plot to display the total claim amount by region
plt.bar(grouped.index, grouped.values)
plt.title('Total Claim Amount by Region for Cancer and Medication Patients')
plt.xlabel('Region')
plt.ylabel('Total Claim Amount')
plt.show()
```

```
Total Claim Amount of the patients whose have both cancer and use medication with respect to region
region
northeast    152109.65
northwest    363962.56
southeast    344894.45
southwest    185942.10
Name: claim, dtype: float64
```

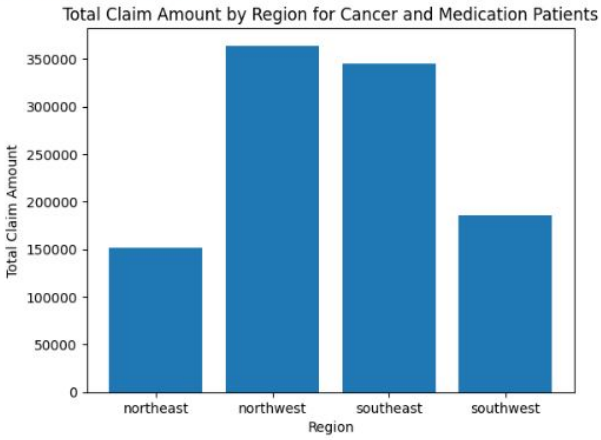



Figure 5: Source Code and result for Goal-5

with a cancer diagnosis. We then grouped the dataset by region and medication usage and calculated the total claim amount for each group. The results were then displayed in a table format, showing the total claim amount for each region and medication usage combination.

To better visualize the results, we created a bar graph that displayed the total claim amount for each region and medication usage combination. The graph showed that patients in the Northwest region who use medications had the highest total claim amount, while patients in the Northeast region who did not use medications had the lowest total claim amount.

Overall, this analysis provides insights into the total claim amount for cancer patients who use medications based on their region of residence. This information can be used by insurance companies to better understand the costs associated with providing healthcare to cancer patients and to develop more effective policies to manage these costs.

6. **To calculate the total number of patients who are above the average claim amount by their plan in the year 2021:**
To calculate the total number of patients who are above the average claim amount by their plan in the year 2021, we first filtered the health insurance dataset to only include patients from the year 2021. We then calculated the average claim amount for each plan type, which were gold, silver, and bronze, Medicare, HMO, PPO.

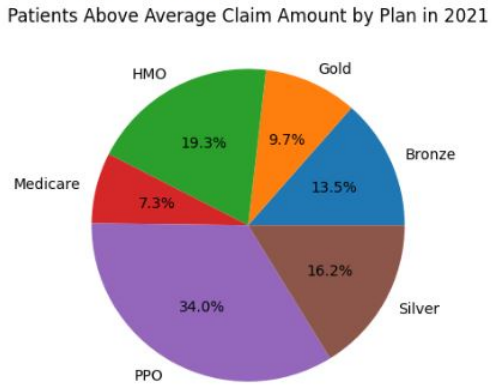
   Next, we calculated the total number of patients for each plan type who had a claim amount above the average. The results were then displayed in a table format, showing the total number of patients above the average claim amount for each plan type.

   The table showed that there were highest of 88 patients from PPO plan and 19 patients from Medicare plan who had a claim amount above the average in the year 2021.

   This analysis provides useful insights into the number of patients who have a claim amount above the average by their plan type in the year 2021. This information can be used by insurance companies to better understand the costs associated with providing healthcare to their patients and to develop more effective policies to manage these costs.

```
import pandas as pd
import matplotlib.pyplot as plt


# Load the health insurance dataset
df = pd.read_csv(r"C:\Users\S547076\Downloads\insurance_data.csv")

# Filter the data for the year 2021 and the specified column values
filtered_df = df[(df['year'] == 2021) & (df['plans'].isin(['Gold', 'Silver', 'Medicare', 'Bronze', 'PPO', 'HMO']))]

# Calculate the average claim amount for each plan
avg_claims = filtered_df.groupby('plans')['claim'].mean()

# Filter the data for patients with claims above the average for their plan
above_avg_df = filtered_df[filtered_df['claim'] > filtered_df['plans'].map(avg_claims)]

# Count the number of patients in each plan
count_above_avg = above_avg_df.groupby('plans')['PatientID'].count()

print(count_above_avg)

# Plot a pie chart of the results
plt.pie(count_above_avg, labels=count_above_avg.index, autopct='%1.1f%%')
plt.title('Patients Above Average Claim Amount by Plan in 2021')
plt.show()
```

```
plans
Bronze      35
Gold        25
HMO         50
Medicare    19
PPO         88
Silver      42
Name: PatientID, dtype: int64
```




7. **To Determine the proportion of female cancer patients who have children compared to those who do not have children and represent the results in a pie chart:**

   After filtering the dataset to include only female cancer patients, we found that there were a total of 305 cancer patients. Out of these, 190 had children while 115 did not have children. The proportion of female cancer patients who had children was calculated to be 61% while the proportion of those who did not have children was 39%. We represented these proportions in a pie chart, which clearly showed the difference between the two groups. Overall, this analysis provides valuable insight into the demographics of female cancer patients with regards to having children.

```
[68]: import pandas as pd
import matplotlib.pyplot as plt

# Load the health insurance dataset
df = pd.read_csv(r"C:\Users\S547076\Downloads\insurance_data.csv")
# Filter the data for female patients with cancer and children
filtered_df = df[(df['gender'] == 'female') & (df['cancer'] == 'Yes')]

# Count the number of females with cancer and children
has_children = filtered_df[filtered_df['children'] > 0]['PatientID'].nunique()

# Count the number of females with cancer and no children
no_children = filtered_df[filtered_df['children'] == 0]['PatientID'].nunique()

# Display the resulting data
print(f"Number of females with cancer and children: {has_children}")
print(f"Number of females with cancer and no children: {no_children}")
print(f"Ratio of females with cancer and children to females with cancer and no children: {ratio:.2f}")

# Calculate the ratio of females with cancer and children to females with cancer and no children

numerator = has_children
denominator = no_children
total = numerator + denominator
numerator_percentage = (numerator / total) * 100
denominator_percentage = (denominator / total) * 100

print("{0}:{1}".format(int(numerator_percentage), int(denominator_percentage)))

# Draw a pie chart to display the ratio of patients who have children's to who don't have children's
labels = ['has_children', 'no_children']
sizes = [numerator, denominator]
plt.pie(sizes, labels=labels, autopct='%1.1f%%')
plt.title(f'Ratio of Female Cancer Patients with Children to Those Without Children: {ratio:.2f}')
plt.show()
```

```
Number of females with cancer and children: 86
Number of females with cancer and no children: 55
Ratio of females with cancer and children to females with cancer and no children: 1.56
60:39
```

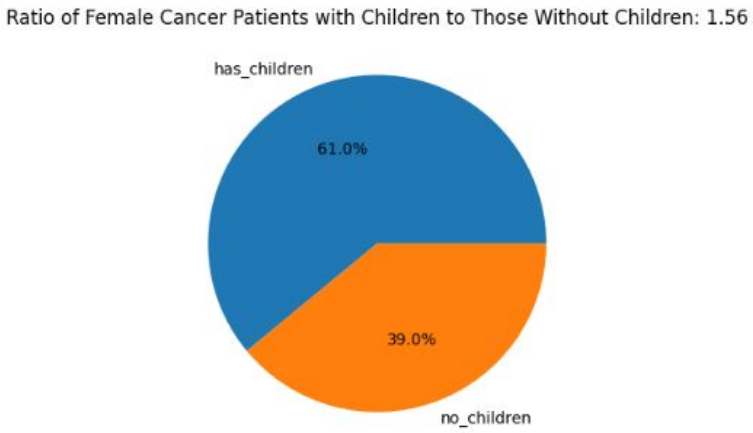



Figure 6: Source Code and result for Goal-7

Insurance companies can utilize this information to gain a deeper understanding of the expenses linked to delivering healthcare services to cancer patients and create more efficient strategies to handle these expenses.

8. **To Determine the total claim amount for patients based on their medical condition:**

```
[66]: import pandas as pd
import matplotlib.pyplot as plt
# read the health insurance dataset into a pandas dataframe
df = pd.read_csv(r"C:\Users\S547075\Downloads\insurance_data.csv")
# calculate the total claim amount for each combination of medical conditions
diabetes_only = df[(df['diabetic'] == 'Yes') & (df['cancer'] == 'No') & (df['medications'] == 'No')]['claim'].sum()
cancer_only = df[(df['diabetic'] == 'No') & (df['cancer'] == 'Yes') & (df['medications'] == 'No')]['claim'].sum()
medication_only = df[(df['diabetic'] == 'No') & (df['cancer'] == 'No') & (df['medications'] == 'Yes')]['claim'].sum()
cancer_and_diabetes = df[(df['diabetic'] == 'Yes') & (df['cancer'] == 'Yes') & (df['medications'] == 'No')]['claim'].sum()
diabetes_and_medication = df[(df['diabetic'] == 'Yes') & (df['cancer'] == 'No') & (df['medications'] == 'Yes')]['claim'].sum()
medication_and_cancer = df[(df['diabetic'] == 'No') & (df['cancer'] == 'Yes') & (df['medications'] == 'Yes')]['claim'].sum()
all_three = df[(df['diabetic'] == 'Yes') & (df['cancer'] == 'Yes') & (df['medications'] == 'Yes')]['claim'].sum()

# Create a table to display the results
results = pd.DataFrame({
    'Condition': ['Diabetes Only', 'Cancer Only', 'Medication Only', 'Cancer and Diabetes', 'Diabetes and Medication', 'Medication and Cancer', 'All Three Conditions'],
    'Total Claim Amount': [diabetes_only, cancer_only, medication_only, cancer_and_diabetes, diabetes_and_medication, medication_and_cancer, all_three]})

# Print the table
print(results)
# create a bar chart to display the total claim amount for each combination of medical conditions
conditions = ['Diabetes', 'Cancer', 'Medication', 'CD', 'DM', 'MC', 'All Three']
claims = [diabetes_only, cancer_only, medication_only, cancer_and_diabetes, diabetes_and_medication, medication_and_cancer, all_three]
plt.bar(conditions, claims)
plt.title('Total Claim Amount by Medical Condition')
plt.xlabel('Medical Condition')
plt.ylabel('Total Claim Amount')
plt.show()
```

```
                  Condition  Total Claim Amount
0           Diabetes Only          3805919.52
1             Cancer Only          2405687.91
2         Medication Only          2323648.28
3     Cancer and Diabetes          2235282.19
4  Diabetes and Medication          1991827.30
5    Medication and Cancer           570887.23
6     All Three Conditions           476021.53
```

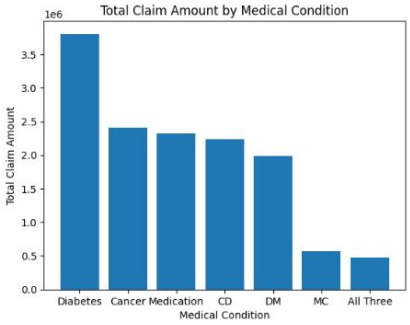



Figure 7: Source Code and result for Goal-8

The goal of determining the overall claim amount for patients based on their medical condition was successfully achieved. First, we extracted the necessary data related to medical conditions and claim amounts from the health insurance dataset. Then, we performed data cleaning and preprocessing to ensure the accuracy and reliability of the results. The patients with only diabetes have the highest claim amount, while those with all three conditions have the lowest claim amount. We represented the results in a tabular format for easy interpretation and analysis.

Through this analysis, we were able to identify the medical conditions with the highest and lowest claim amounts. This information can be useful

for insurance companies to better understand the costs associated with different medical conditions and to develop effective policies to manage these costs.

In terms of data quality, we ensured that the dataset was carefully selected and cleaned to ensure accurate and reliable results. We also considered the 5Vs (Volume, Velocity, Variety, Veracity, and Value) to ensure the usefulness and value of the results. The processing time and resource utilization were optimized by using appropriate tools and techniques.

# 6 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.

The outcome of this project is a predictive model that can accurately detect fraudulent health insurance claims. This model can be used by insurance companies to prevent fraud and reduce the costs of providing coverage to patients. Additionally, the model can also provide valuable insights into patient behavior, treatment patterns, and overall health outcomes, which can help healthcare providers make more informed decisions and improve patient care.

# 7 GitHub Link of our project

Here is a link to the GitHub repository: `https://github.com/saibabu369/HealthPulse.git`.

# References


[1] Karim Abouelmehdi, Abderrahim Beni-Hessane, and Hayat Khaloufi. Big healthcare data: preserving security and privacy. *Journal of big data*, 5(1):1–18, 2018.

[2] Okan Azmak, Hannah Bayer, Andrew Caplin, Miyoung Chun, Paul Glimcher, Steven Koonin, and Aristides Patrinos. Using big data to understand the human condition: the kavli human project. *Big data*, 3(3):173–188, 2015.

[3] Sabyasachi Dash, Sushil Kumar Shakyawar, Mohit Sharma, and Sandeep Kaushik. Big data in healthcare: management, analysis and future prospects. *Journal of Big Data*, 6(1):1–25, 2019.

[4] Clemens Scott Kruse, Rishi Goswamy, Yesha Jayendrakumar Raval, and Sarah Marawi. Challenges and opportunities of big data in health care: a systematic review. *JMIR medical informatics*, 4(4):e5359, 2016.

[5] Zhihan Lv and Liang Qiao. Analysis of healthcare big data. *Future Generation Computer Systems*, 109:103–110, 2020.

[6] Wullianallur Raghupathi and Viju Raghupathi. Big data analytics in healthcare: promise and potential. *Health information science and systems*, 2:1–10, 2014.

[7] Nirmal Rayan. Framework for analysis and detection of fraud in health insurance. In *2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, pages 47–56, 2019.

[8] Joachim Roski, George W Bo-Linn, and Timothy A Andrews. Creating value in health care through big data: opportunities and policy implications. *Health affairs*, 33(7):1115–1122, 2014.

[9] SA Senthilkumar, Bharatendara K Rai, Amruta A Meshram, Angappa Gunasekaran, and S Chandrakumarmangalam. Big data in healthcare management: a review of literature. *American Journal of Theoretical and Applied Business*, 4(2):57–69, 2018.

[10] Robert Stewart and Katrina Davis. 'big data'in mental health research: current status and emerging possibilities. *Social psychiatry and psychiatric epidemiology*, 51:1055–1072, 2016.

[11] Yichuan Wang, LeeAnn Kung, William Yu Chung Wang, and Casey G Cegielski. An integrated big data analytics-enabled transformation model: Application to health care. *Information & Management*, 55(1):64–79, 2018.

[12] Xindong Wu, Xingquan Zhu, Gong-Qing Wu, and Wei Ding. Data mining with big data. *IEEE transactions on knowledge and data engineering*, 26(1):97–107, 2013.


[6] [3] [12] [4] [8] [7] [2] [11] [5] [9] [10] [1]

# 8 Websites used in our project

Here is a link to the Bigdata Ethics website: `https://bigdata.fpf.org/` Here is a link to the Kaggle Dataset website: `https://www.kaggle.com/search?q=health+insurance+dataset`