Big Data Management
(Final Report)

Assessment 1.2

Student ID: 23206188

Student Name: Dinesh Thapa

Part of a

Big Data Management - CMP7203

BIRMINGHAM CITY UNIVERSITY

FACULTY OF COMPUTING ENGINEERING AND THE BUILT ENVIRONMENT

Table of Contents

Table of Figures

List of Tables

1 Introduction

Big Data has evolved from the discipline of statistical analysis to sophisticated data platform technologies (Yaqoob *et al.*, 2016). Big Data refers to extremely large and diverse collections of structured, semi-structured, and unstructured data that organizations collect, store, process, and analyze it for valuable insights and information (Sagiroglu and Sinanc, 2013).

Mainly Big Data is often characterized by three V's:

Volume (The enormous amount of data generated from various sources),

Variety (The various formats and types of data), and

Velocity (The speed at which data is generated and needs to be processed in real-time)

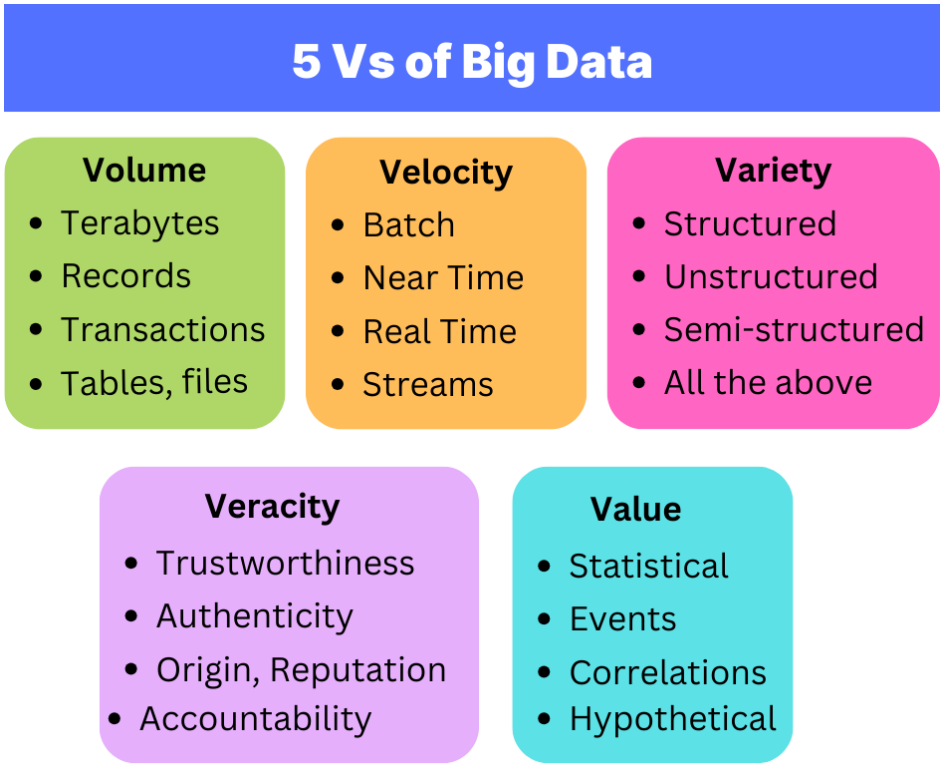


Figure 1: 5 Vs of Big Data

The three fundamental features of big data were first proposed by Doug Laney (Volk *et al.*, n.d.). In addition to these original three Vs, there are other two Vs that are popular and defines the power of Big Data; they are Veracity and Value. Veracity or

Variability, which denotes the reliability of the data, was introduced by IBM and Microsoft as the fourth V to characterize big data (Khan *et al.*, 2018). Furthermore, as an additional V in the definition of big data, McKinsey Co. added Value that means how big data can provide real business value or benefits (Al-Sai *et al.*, 2019). Organizations always need to confirm that data is relevant to their business issues before it is used for big data analytics (Vassakis *et al.*, 2018).

2 Big Data Processing Paradigms

Big data processing paradigms are fundamental approaches and technologies designed to deal with particular challenges provided by large-scale datasets which are defined by their enormous volume, different data types, and high velocity of data generation (Casado and Younas, 2015). Big data needs innovative methods, tools, and techniques for solving emerging problems associated with these particular attributes (Pathak *et al.*, 2020).

The figure is derived from ("Processing paradigms. | Download Scientific Diagram", n.d.)

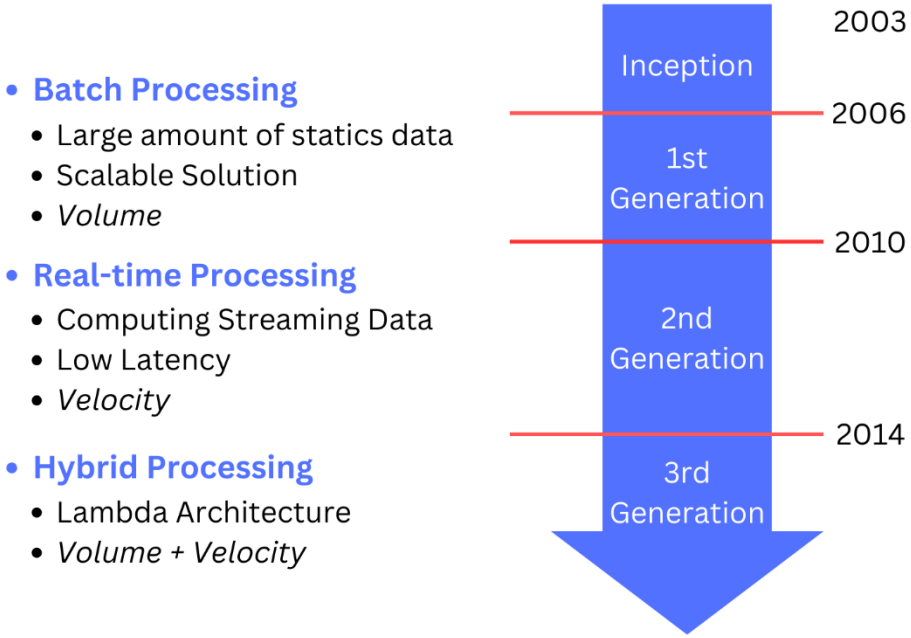


Figure 2: Processing Paradigms in Big Data

According to the literature (Sinanc Terzi *et al.*, 2016), the major big data processing paradigms are classified based on three Vs: volume, velocity, and variety; which are batch, real-time and hybrid processing as shown on Figure 2.

2.1 Batch Processing

Batch processing is the process of dividing a huge volume of data into discrete chunks or blocks and processing each block separately and sequentially over time (Benjelloun *et al.*, 2020). This type of processing is often used on data that has already been collected and stored over time such as analyzing all transactions made by a banking institution in a month (Zheng *et al.*, 2019).

The figure is derived from (https://datascience.aero/batch-processing-streaming-processing-aviation/)

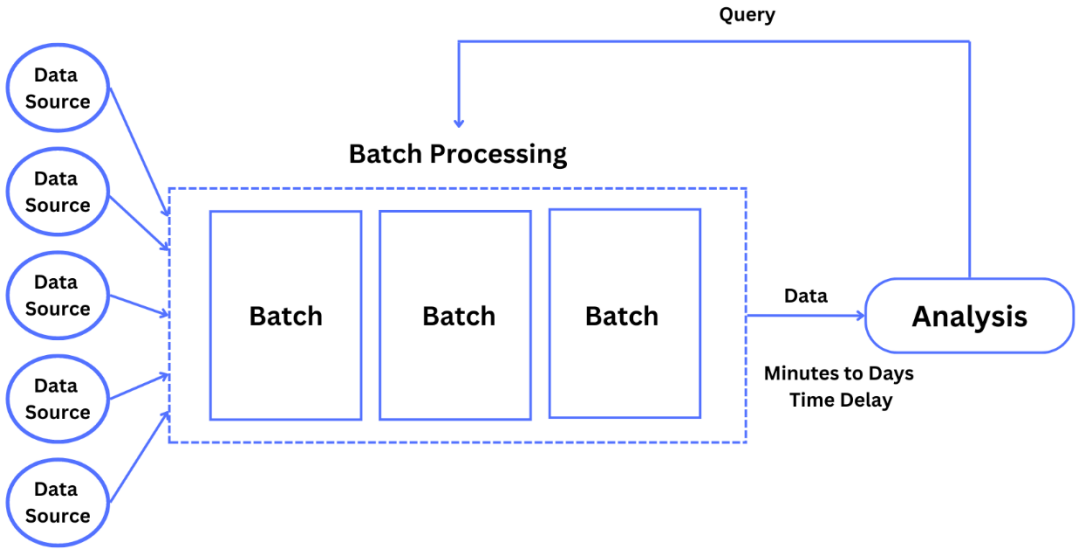


Figure 3: Batch Processing Working Mechanism

The primary feature of batch processing is that jobs executed in batches, starting and finishing independently of real-time data streams. Batch processing jobs are often conducted simultaneously but in a specific order dividing large tasks into smaller parts for improved efficiency (Goudarzi, 2019).

One of the key benefits of batch processing is its ability to operate offline which reduces resource usage and processor usability (Meehan *et al.*, 2016). Different tools available for batch processing are: Apache Hadoop, Apache Spark, Apache Flink, and Amazon EMR (Elastic MapReduce).

An example of batch processing is to assess the visitors weblog file of a website in order to know customers purchasing habits based on their interactions and activities in the website.

Apache Hadoop for Batch Processing

Apache Hadoop is a widely used open-source system for batch processing and distributed computing using a method called MapReduce (Nandimath *et al.*, 2013). It consists of master/slave architecture in which single NameNode work as a master and multiple DataNodes work as a slave. It stores data in a distributed filesystem known as Hadoop Distributed File System (HDFS). This data is divided into smaller sections called splits which are the building blocks for MapReduce processing (Greeshma and Pradeepini, 2016).

Figure Source: ("Hadoop framework The Yet Another Resource Negotiator (YARN), which aids... | Download Scientific Diagram", n.d.)

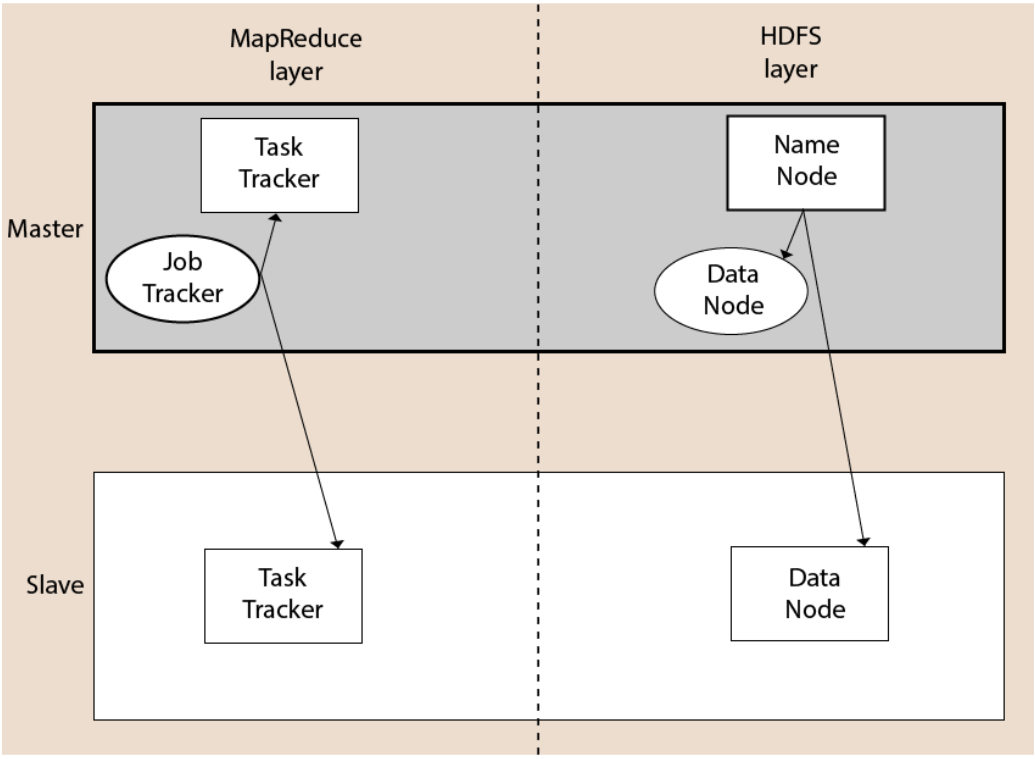


Figure 4: Hadoop Architecture

In MapReduce, the processing begins with a "map" task that reads these splits and applies a map function to each producing intermediate results (Dittrich and Quiané-Ruiz, 2012). These results are then consolidated by a "reduce" task which combines and refines them into final output files. MapReduce offers several advantages as it simplifies distributed programming, delivers almost linear speed improvements with more computing resources, scales well, and maintains fault tolerance (Ghazi and Gangodkar, 2015).

2.2 Real Time/Stream Processing

Real-time processing involves processing data almost instantly and data must continuously enter which will show real-time insights in the system (Liu *et al.*, 2014). When raw data is received in the system, it is instantly processed to provide near-instant decision-making (Safaei, 2017). Instead of being stored for future purpose, it is made available to provide insights as quickly as possible supporting organizations' profitability, efficiency, and business outcomes.

The figure is derived from (https://datascience.aero/batch-processing-streaming-processing-aviation/)

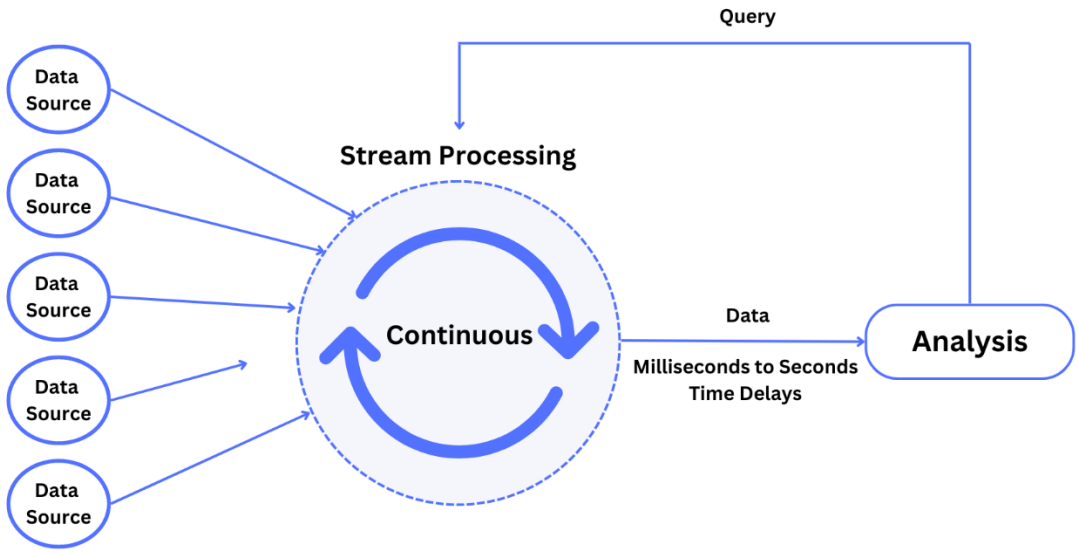


Figure 5:Stream Processing Working Mechanism

The fastest data processing method that processes data quickly and yields the most accurate results is real-time data processing (Gurcan and Berigel, 2018). For instance, a real-time traffic monitoring system like Google Maps gathers data in real-time to display traffic jams and can initiate traffic management systems like high-occupancy lanes automatically (D'Alconzo *et al.*, 2019). Google updates its maps dynamically by gathering that data in real-time. Batch processing is slower and less accurate than real-time processing which is often referred to as stream data processing.

Real-time data processing has many benefits for businesses (Ciu *et al.*, 2007). First, it allows companies to understand data right away and make decisions based on what is happening at that moment. This helps them to be more flexible and reduces the

chances of making mistakes. Second, real-time processing can help detect problems in the system quickly, like fraud or rule-breaking, so action can be taken fast.

Different tools available for Real-time processing are Apache Spark, Apache Kafka, Apache Flink, Amazon Kinesis, Azure Stream Analytics, and many more.

Several application areas of real-time processing include transportation data processing, energy supply optimization and garbage management in smart cities, analyzing streaming data from gaming platform.

Apache Kafka for Real-Time Processing

Apache Kafka is an open-source distributed real-time or streaming platform that transforms how applications manage and process data streams. It is a powerful tool for managing real-time data feeds and is designed to be highly scalable, available, fault-tolerant and superior tool for massive scale data processing job.

Figure Source: https://blog.devgenius.io/consume-the-same-data-with-different-consumer-groups-in-kafka-spring-boot-e7ba8cce31df

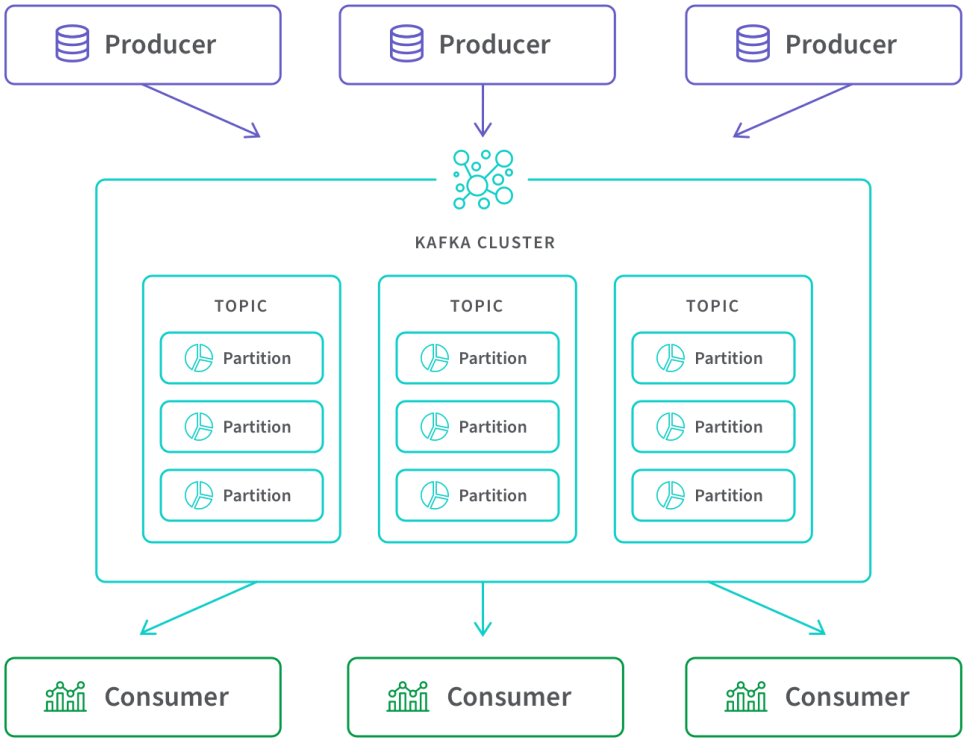


Figure 6: Apache Kafka Architecture

Producer: An application that sends events or data into specific topics within Kafka.

Cluster: A group of one or more servers (called brokers).

Topic: A way to categorize and permanently store events or data.

Partition: A method to spread out data across multiple servers (brokers).

Consumers: Applications that retrieve and process events or data from Kafka partitions.

The major benefits of Kafka are scalability, speed, and durability.

2.3 Hybrid Processing

Hybrid Big Data Processing refers to an approach that combines the strengths of both batch processing and real-time processing to address the diverse needs of business requirements which often demand simultaneous handling of historical and real-time data (Londhe and Prasada Rao, 2018). Hybrid data processing can efficiently handle large volumes of data in scheduled intervals and real-time processing which provides low-latency, immediate data insights (Dos Anjos *et al.*, 2020).

Hybrid Processing offers flexibility by allowing organizations to tailor their data processing strategies based on specific use cases (Pishgoo *et al.*, 2021). By combining batch and real-time processing, hybrid models optimize resource utilization. Historical data can be processed in batch mode during off-peak hours utilizing cost-effective computing resources.

In summary, hybrid big data processing integrates batch and real-time processing capabilities to address the multifaceted nature of business data requirements. By combining the strengths of both paradigms, organizations can achieve a balance between historical data analysis and real-time decision-making, thereby enhancing overall data-driven operations (Serhani *et al.*, 2016).

Lambda Architecture for Hybrid Processing

The figure is sourced from https://tinalekse.substack.com/p/lambda-kappa-and-delta-architectures

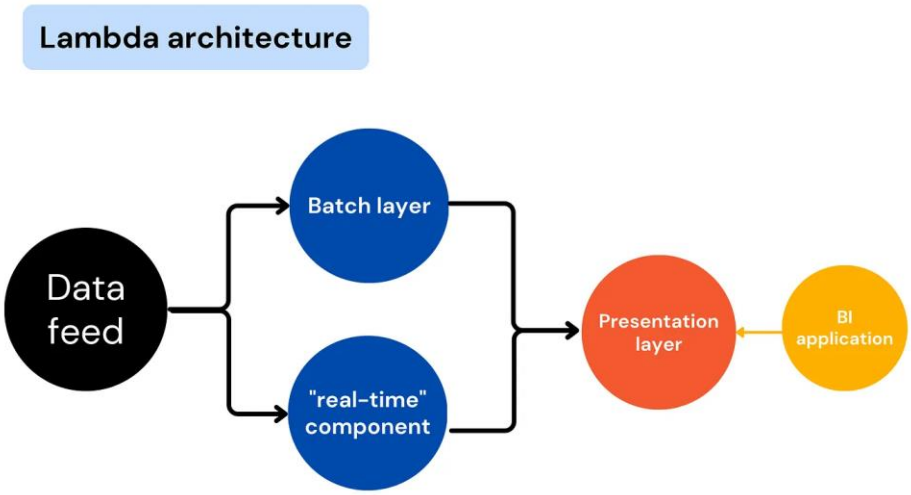


Figure 7: Lambda Architecture

In Lambda architecture, data processing can happen in two different ways: batch processing and real-time streaming (Kiran *et al.*, 2015). This means that data is handled either in chunks (batch) or as it comes in (real-time). These processes use big systems like Hadoop for batch processing and Apache Kafka for real-time streaming. The data is usually stored across many computers in a system like HDFS or Apache Cassandra. The results from both are put together to give out the final data. Lambda architecture is more complicated compared to Kappa and Delta architectures (Hasani *et al.*, n.d.).

2.4 Other Data Processing Architectures

Kappa Architecture

The figure is sourced from https://tinalekse.substack.com/p/lambda-kappa-and-delta-architectures

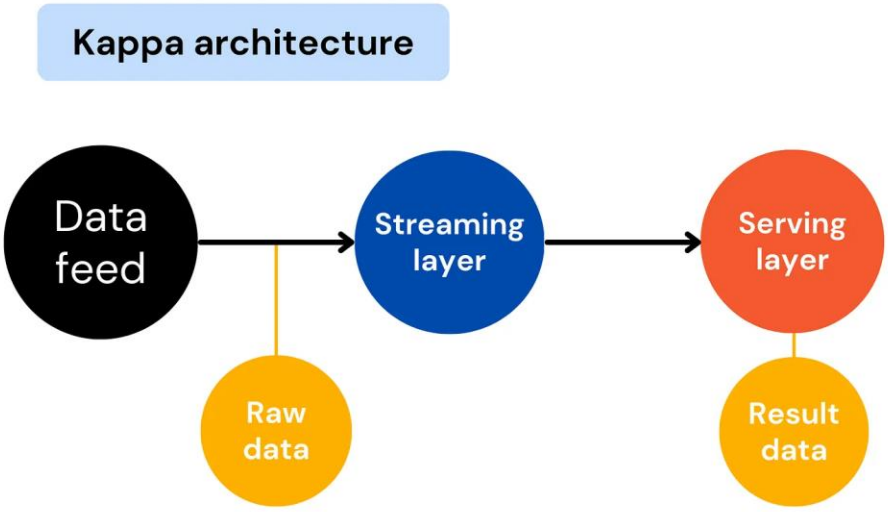


Figure 8: Kappa Architecture

In Kappa Architecture, data is handled in one streamlined process instead of separate batch and real-time pipelines (Singh *et al.*, 2019). This approach uses a streaming framework exclusively where, data flows continuously. Typically, data is stored in a distributed system like HDFS (Hadoop Distributed File System).

Delta Architecture

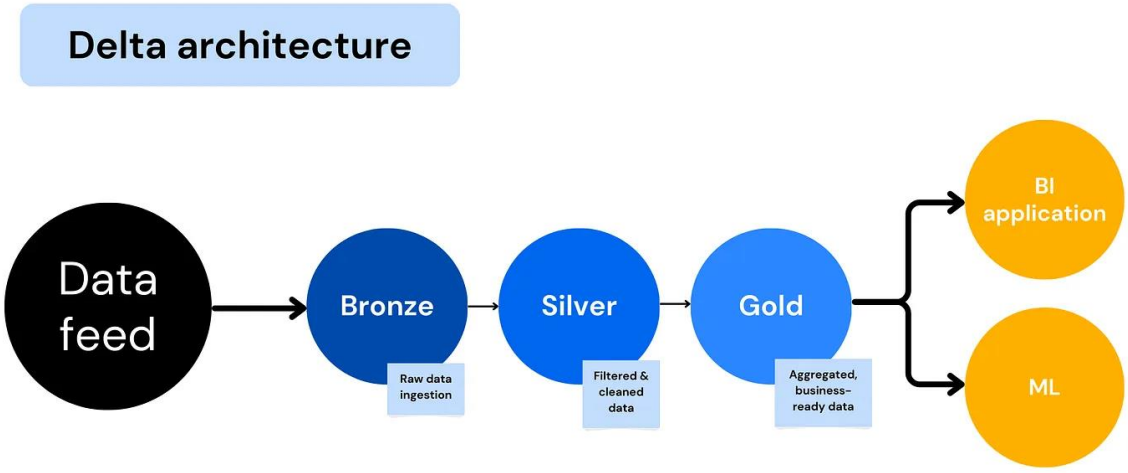


Figure 9: Delta Architecture

The figure is sourced from https://tinalekse.substack.com/p/lambda-kappa-and-delta-architectures

In Delta Architecture, data is managed through three layers: batch processing, speed processing, and serving layers. This means data is handled in chunks as it comes in and when it's ready to be served or used.

It uses a mix of batch-processing tools like Hadoop, real-time streaming engines like Kafka, and databases that can handle data in real-time. Data is usually stored in a data lake or a cloud-based storage system for flexibility and scalability (L'Esteve, 2023).

This architecture is designed to overcome the limitations of both Kappa and Lambda architectures providing a more flexible and better way to handle different types of data processing needs (Ait Errami *et al.*, 2023).

3 Big Data Paradigms in Various Organizations & Use Cases

Big Data has completely changed the way that analytics and data management work providing organizations unprecedented opportunities to extract insightful information from huge and complex databases. Organizations are implementing different Big Data paradigms based on their requirements and nature of problem domain.

Below are lists of different organizations along with the Big Data processing paradigms they are using and their corresponding use cases.

Table 1: Big Data Paradigms in Various Organizations & Use Cases

S.N.	Organization	Big Data Paradigm Implemented	Use Case/Scenario
1	UPS	Real-time Processing	Optimizing Delivery Routes, Reducing Fuel Consumption, Improving Operational Efficiency
2	United Healthcare	Batch Processing, Hybrid Processing	Customer Satisfaction Analysis, Real-time Interventions, Data Storage and Processing, Integration with Statistical Analysis
3	Bank of America	Batch Processing	Customer Segment Analysis, Fraud Detection and Prevention, Customer Insights and Trend Analysis
4	Sears	Real-Time Processing	Real-Time Data Acquisition, Immediate Measurement and Response, Accelerated Marketing Campaigns Analysis

5	Facebook	Batch Processing	Large Scale Parallel Processing with Hadoop, Historical Data Analysis for Product Improvements, Data Loading and Processing Volumes
6	Amazon	Batch Processing, Real-Time Processing, Hybrid Processing	Amazon EMR - Large Data Analytics, Amazon Kinesis - real-time analytics, Data Loading and Processing Volumes
7	LinkedIn	Batch Processing, Real-Time Processing	Social Graph Computation, OLAP and Real-time Queries, Data Warehousing and Storage
8	Capital One	Real Time Processing	Real-Time Offer Optimization, Customer Segmentation and Targeting, Campaign Performance Analysis
9	General Electric (GE)	Real Time Processing	Sensor Data for Real-Time Insights, Immediate Decision-Making, Operational Optimization
10	Miniclip	Hybrid Data Processing	Customer Experience Monitoring, Customer Retention, Iterative Product Improvement
11	Netflix	Real-Time Processing	Real-Time Insights for Content Commissioning, Dynamic Content Selection, Immediate Decision-Making
12	Starbucks	Hybrid Data Processing	Customer Insights, Personalization, Immediate Marketing and Engagement
13	Spotify	Real-Time Processing	Dynamic Content Delivery, Streaming Analytics, Optimizing Service Performance
14	American Express	Real-Time Processing	Fraud Detection in Real-Time, Machine Learning Applications, Immediate Decision-Making
15	Apple	Batch Processing, Real-Time Processing	App Usage Analytics and Future Designs, Health Monitoring with Wearable Technology
16	McDonald's	Real-Time Processing	Customer Experience Enhancement, Operational Efficiency
17	Instagram	Real-Time Processing	Personalized Recommendations, Continuous Improvement and Growth, Integration with AI
18	Airbnb	Batch Processing	Enhanced Search Features, Dynamic Pricing Guidance, Customer Experience Optimization
19	Adello	Real-Time Processing	AI-driven Ad Fraud Detection, Automated Decision-Making, Dynamic Audience Targeting
20	Twitter	Batch Processing	Tweet Storage and Processing, Log File Processing, Historical Data Analysis

4 Future Role of Big Data in the Age of Large Language Models

Big Data has the potential to revolutionize multiple industries on how decisions are made and driving innovation (Syed *et al.*, 2013). The potential opportunities and impact of big data are practically limitless as we enter the beginning of a data-driven era (Agrawal *et al.*, 2011).

Large language models (LLMs) are sophisticated artificial intelligence (AI) systems created to understand the nuances of human language and produce creative answers to queries (Bao *et al.*, 2023). Successful LLMs like GPT, Gemini, Falcon, Claude are trained on massive datasets typically measured in petabytes which are sourced from books, articles, journals, websites, and different text-based sources (Han *et al.*, 2021). Large Language Models requires a massive scale of dataset to learn and improve its decision-making processes whereas big data analytics provides platform for better data analysis and processing (Birhane *et al.*, 2023).

The total amount of data created was 64.2 zettabytes in 2020 and by the year 2025, global data creation is projected to grow to more than 180 zettabytes according to ("Data growth worldwide 2010-2025 | Statista", n.d.).

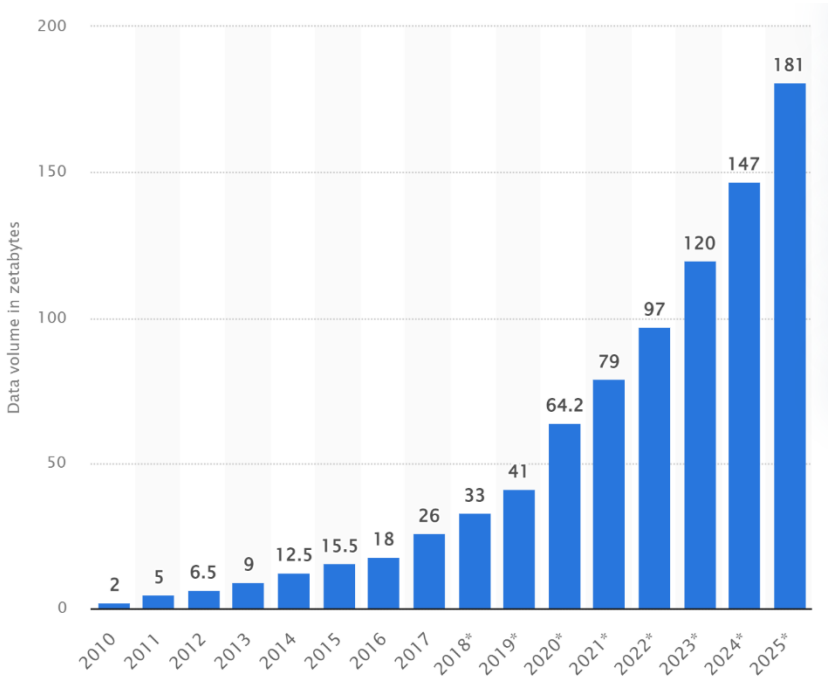


Figure 10: Data Growth Worldwide 2010-2015 (Source: Statista)

With the help of big data, companies can process this huge volume of data they are generating and train their own customized Large Language Models to provide tailored customer support based on their services and products (ZhaoHaiyan *et al.*, 2024). Large Language Models power many new tools such as chatbots for customer care, content creation tools, financial analysis, scientific research, and advanced search tools. But they need big data to train them well and without access to better data, LLM would lack the necessary knowledge to provide better decisions and answers to queries (Ding *et al.*, 2023). This mutually beneficial relationship has produced ground-breaking innovations that are revolutionizing sectors including e-commerce, healthcare, and finance. Examples of these innovations include recommendation systems, natural language processing, and predictive analytics (Wang *et al.*, 2023). We can say that Big Data is the fuel and AI like LLM is the engine that drives innovation. Big Data plays a pivotal role in the development and optimization of LLMs such as GPT (Generative Pre-trained Transformer) models.

The key aspects that emphasize the importance of Big Data for LLMs are: -

a. Training Data

Big Data gives LLMs access to the volume and variety of text corpora they need to master syntactic structures, linguistic patterns and semantic intricacies in a variety of languages and subjects (Chen and Lin, 2014).

b. Generalization and Adaptation

Big Data gives Large Language Model (LLMs) an in-depth understanding of language use which helps them produce well-reasoned, contextually appropriate answers for a variety of subjects and situations (Zhang *et al.*, 2024).

c. Fine-tuning and Optimization

Big Data makes it easier to continuously optimize and fine-tune LLMs by using methods like fine-tuning. Organizations can improve the model's performance and customize it to fit particular application requirements, including sentiment analysis, summarization, or translation by introducing LLMs to new data unique to particular domains or jobs (Malladi *et al.*, 2023).

d. Complex Pattern Recognition

Having access to Big Data allows LLMs to identify sentiment changes, find small semantic correlations, and extract useful information from unstructured text sources enabling a range of applications in knowledge extraction, content creation, and information retrieval (Pal and Pal, 2016).

e. Real-time Adaptation and Feedback

LLMs must react instantly to change user interactions and linguistic trends in dynamic situations. Continuous input gained from Big Data streams helps LLMs to stay up to date with latest information that will help in improving user experience and engagement with applications such as virtual assistants, chatbots, and conversational agents (Chang *et al.*, 2024).

Big Data is essentially the foundation for the creation, improvement, and ethical implementation of large language models. This will drive innovation in a variety of applications and sectors while ensuring ethical and Responsible AI development (Sarker, 2024). Large Language Models like GPT-3.5 are prime examples of the fusion between big data and AI as they rely on vast amounts of data to train and continually improve their language processing capabilities. With the exponential growth of data, LLMs have the potential to completely change the way we interact with information as they have the ability to comprehend, generate, and analyze text at an unprecedented scale that unlocks insights that is hidden within massive datasets that were previously impractical to process manually. From identifying market trends and customer preferences to optimizing operational efficiency and mitigating risks, the integration of big data and LLMs enables organizations to make informed decisions with greater speed and precision.

5 Exploratory Data Analysis (EDA)

Exploratory Data Analysis is a fundamental process used by data scientists to understand a dataset, its structure, and its characteristics. Using different statistical techniques and data visualization, it allows researchers to extract valuable insights from the data before diving into machine learning, analytics, or hypothesis testing tasks (Chatfield, 1986). Implementing summary statistics, different statistical measures, and graphical representations, it helps to find the hidden trends and patterns within the dataset.

Now, Exploratory Data Analysis will be performed to understand the diabetes dataset better which provides the foundation for further analysis. After EDA is done, we do data preprocessing and transformation steps to ensure data quality and reliability. In addition, we will also identify patterns, outliers, skew, anomalies in the data, and any necessary adjustments needed before using this data for machine learning modeling as explained in (Velleman and Hoaglin, 2012).

5.1 Data Description

The dataset below consists of health-related variables and demographic information for individuals that help us to analyze and predict diabetes by training our model on those features. Each row represents a unique individual record with its corresponding feature set.

The dataset column and its description are explained below:

Table 2: Data Description of Diabetes Dataset

S.N.	Column Name	Description
1	Diabetes_binary	Diabetes status: 0 = No diabetes, 1 = Diabetes
2	HighBP	High blood pressure: 0 = No, 1 = Yes
3	HighChol	High cholesterol: 0 = No, 1 = Yes
4	CholCheck	Cholesterol checks in the last 5 years: 0 = No, 1 = Yes
5	BMI	Body Mass Index, numerical value
6	Smoker	Smoked 100+ cigarettes in lifetime: 0 = No, 1 = Yes

7	Stroke	History of stroke: 0 = No, 1 = Yes
8	HeartDiseaseorAttack	History of heart disease or attack: 0 = No, 1 = Yes
9	PhysActivity	Physical activity in past 30 days: 0 = No, 1 = Yes
10	Fruits	Consumes fruit daily: 0 = No, 1 = Yes
11	Veggies	Consumes vegetables daily: 0 = No, 1 = Yes
12	HvyAlcoholConsump	Heavy alcohol consumption: 0 = No, 1 = Yes
13	AnyHealthcare	Has any healthcare coverage: 0 = No, 1 = Yes
14	NoDocbcCost	Needed but couldn't afford doctor visit last year: 0 = No, 1 = Yes
15	GenHlth	General health (1-5, where 1 = excellent, 2 = very good, 3 = good, 4 = fair, and 5 = poor)
16	MentHlth	Number of days with poor mental health in past 30 days
17	PhysHlth	Number of days with poor physical health in past 30 days
18	DiffWalk	Serious difficulty walking/climbing stairs: 0 = No, 1 = Yes
19	Sex	Gender: 0 = Female, 1 = Male
20	Age	Age category (e.g., 1 = 18-24, 9 = 60-64, 13 = 80 or older)
21	Education	Education level (1-6, where 1 = no schooling, 2 = Grades 1 through 8, 3 = Grades 9 through 11, 4 = Grade 12 or GED, 5 = Some college or technical school, 6 = college graduate)
22	Income	Income scale (1-8, where 1 = Less than $10,000, 5 = Less than $35,000, 8 = $75,000 or more)

Source: The dataset description is taken from ("CDC Diabetes Health Indicators - UCI Machine Learning Repository", n.d.) *and it is converted into an easy-to-read format. The provided dataset only had column names. So, this is included to provide more meaning to data.*

Now, based on this dataset, we perform different types of exploratory data analysis to explore the trends, patterns, and major characteristics of the dataset.

Basic Statistics

Figure 11: Basic Statistics

summary	Diabetes_binary	HighBP	HighChol	CholCheck	BMI	Smoker	
count	253680	253680	253680	253680	253680	253680	
mean	0.13933301797540207	0.4290011037527594	0.4241209397666351	0.9626695048880479	28.382363607694735	0.44316855881425415	0
stddev	0.3462943845892145	0.49493446268990027	0.4942098046568846	0.1895707543627261	6.608694201406007	0.4967606667785631	
min	0	0	0	0	12	0	
max	1	1	1	1	98	1	

Figure 11 shows the basic statistics of the dataset columns. Exploring these complete dataset columns' statistics provides us with an overview of the dataset values in each column.

Skewness Analysis

```
Skewness of BMI: 2.12

Kurtosis of BMI: 11.00

Skewness of Age: -0.36

Kurtosis of Age: -0.58

Skewness of PhysHlth: 2.21

Kurtosis of PhysHlth: 3.50

Skewness of MentHlth: 2.72

Kurtosis of MentHlth: 6.44
```

As BMI is positively skewed, it informs that more people have a BMI lower than the mean BMI. Similarly, Physical Health and Mental Health are also positively skewed but Age is slightly negatively skewed which means distribution is not heavily skewed.

1. **Is there a pattern of any missing data or null values in the dataset?**

No, there are no any missing or null values existed in the dataset.

Figure 12: No Null Values

Diabetes_binary	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDiseaseorAttack	PhysActivity	Fruits	Veggies
0	0	0	0	0	0	0	0	0	0	0

2. **What is the distribution of the target variable?**

Figure 13: Distribution of the Diabetes_binary variable

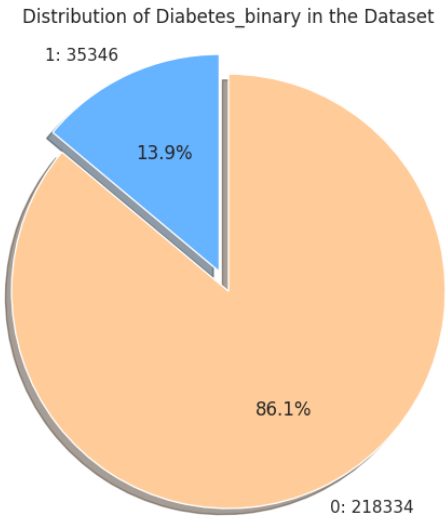


Figure 13 depicts the distribution of the target variable (Diabetes_binary) in which 13.9% is diabetic records whereas 86.1% is non-diabetic records which shows the significant imbalance in class distribution.

3. **How does BMI relate to diabetes prevalence?**

Figure 14: BMI relate to diabetes prevalence

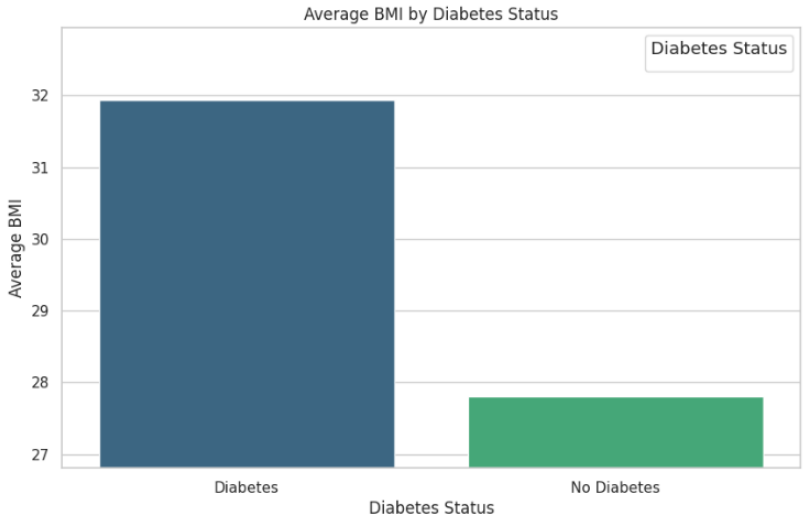


Figure 14 shows that individuals with diabetes have a significantly higher average BMI than those without diabetes. This suggests that higher BMI is associated with increased prevalence of diabetes. Thus, BMI is the most important feature in predicting diabetes.

What is the distribution of BMI across the dataset?

Figure 15: Distribution of BMI across the dataset

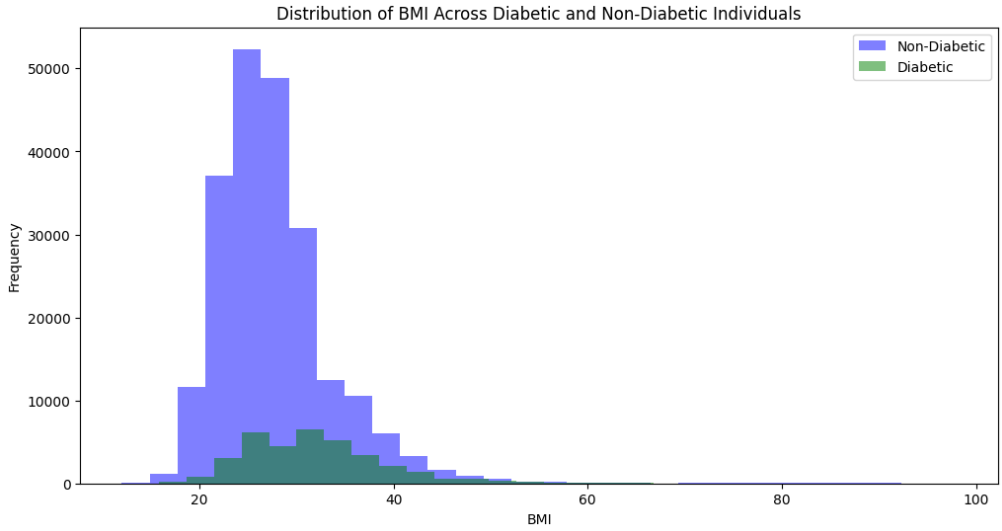


Figure 15 demonstrates that the distribution of BMI is mostly around the 20 to 40 range for both diabetic and non-diabetic individuals. However, diabetic individuals tend to have a higher range of BMI values. It clarifies that BMI is more common among diabetic people than non-diabetic people.

4. **Does the presence of High Blood Pressure (HighBP) and High Cholesterol (HighChol) correlate with increased diabetes risk?**

Figure 16: Correlation between HighBP and HighChol with Diabetes

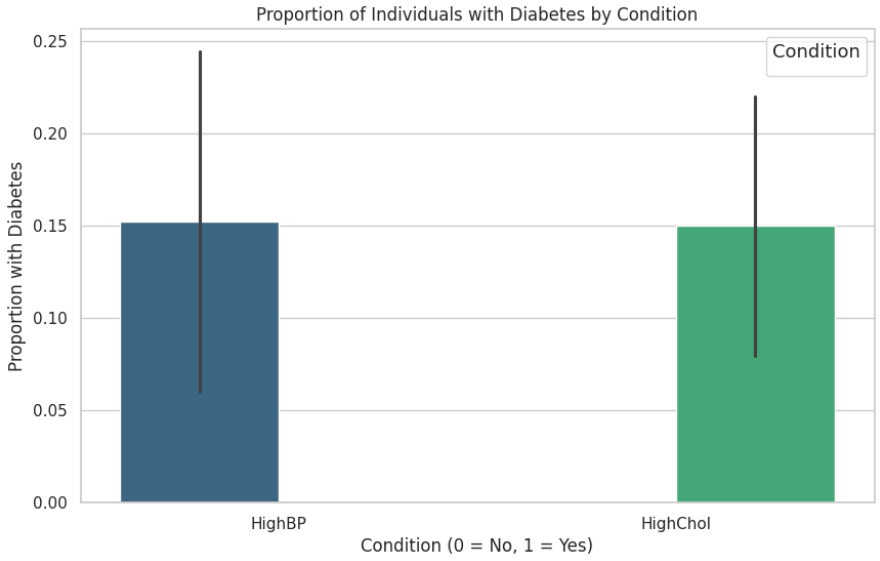


Figure 17: Correlation Table Data Points

HighBP	Proportion_with_Diabetes	Condition
1	0.24445690027474296	HighBP
0	0.06035167171783419	HighBP
1	0.22014852543428354	HighChol
0	0.07981435973961079	HighChol

Figure 16 illustrates that individuals with HighBP show a high proportion of diabetes at about 24.45% whereas those without HighBP have a lower proportion at about 6.04% which is shown in Figure 17. In addition, individuals with HighChol also have a higher proportion of diabetes at about 22.01%.

5. **What are the correlations between features?**

Figure 18: Correlations between features

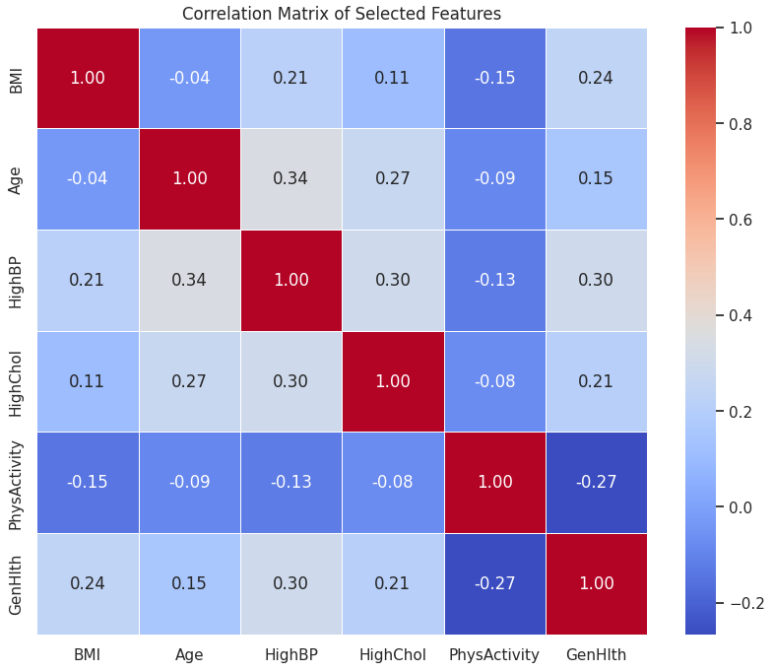


Figure 18 shows that HighBP, HighChol, GenHlth, and Age are positively correlated which means both HighBP and HighChol increase as age increases and it is associated with worse general health.

6. **How are age categories distributed among individuals with and without diabetes?**

Figure 19: Distribution of Age Categories with and without Diabetes

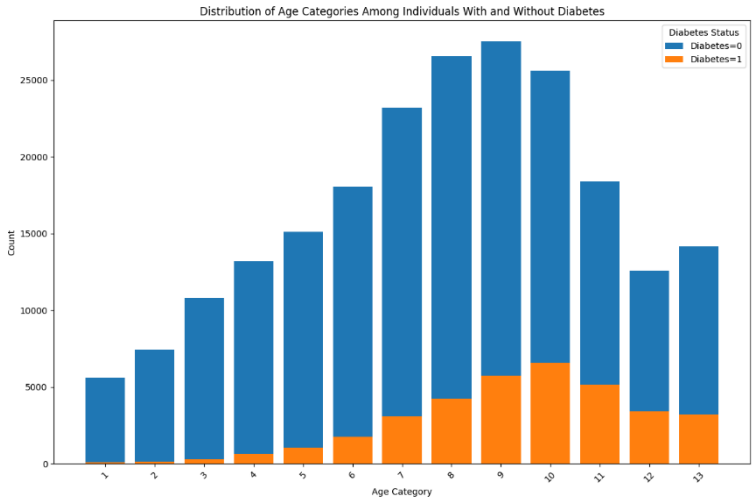


Figure 19 depicts that as age increases, the proportion of individuals with diabetes also increases reaching to peak in the ranges around category 7 to 9.

7. Are there any patterns between different factors such as smoking, alcohol consumption, physical activity, and diabetes?

Figure 20: Patterns between different lifestyle factors and diabetes

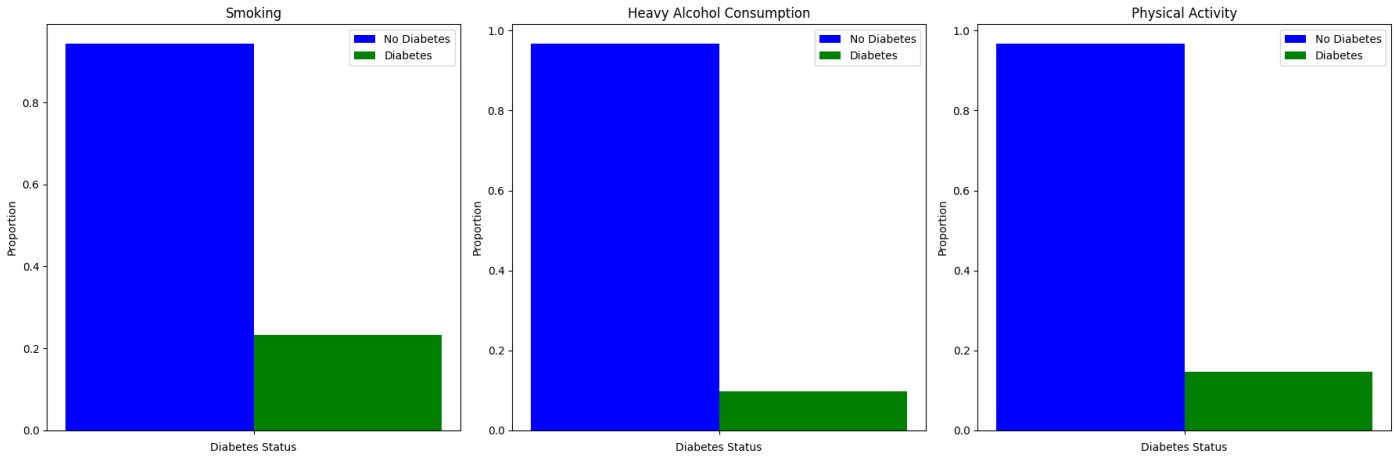


Figure 20 demonstrates that Smoking and Alcohol consumption are less prevalent in diabetes. However, lower physical activity is associated with the highest diabetes prevalence.

8. What is the proportion of general health (GenHlth) with diabetes and education level with diabetes?

Figure 22: General Health with Diabetes

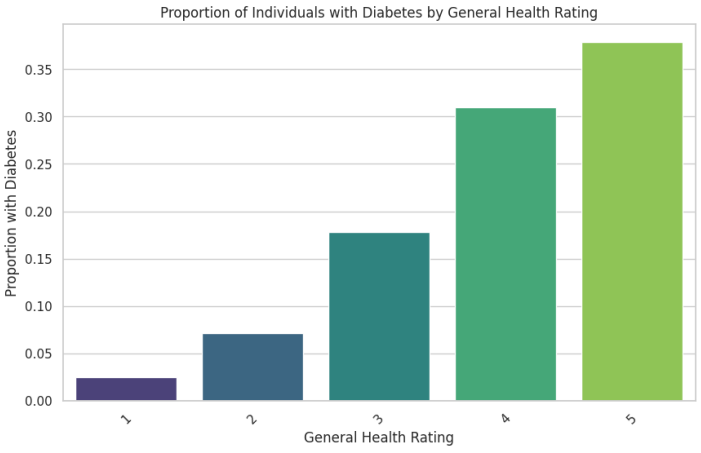


Figure 21: Education Level with Diabetes

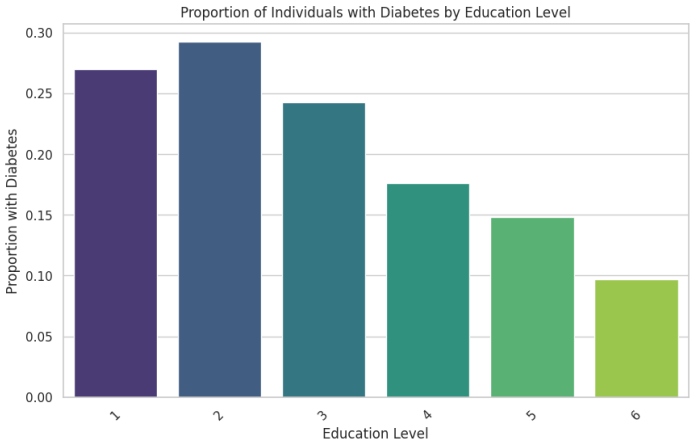


Figure 22 (Left) shows that the proportion of diabetes increases as the general health rating decreases. It means that individuals having poor health ratings (4,5) have a higher proportion of diabetes.

Figure 21 (Right) shows that the proportion of diabetes decreases as the education level increases. It means that individuals having higher education levels (5,6) show a lower proportion of diabetes.

9. How does physical activity impact the risk of diabetes?

Figure 23: Proportion of physical activity with diabetes

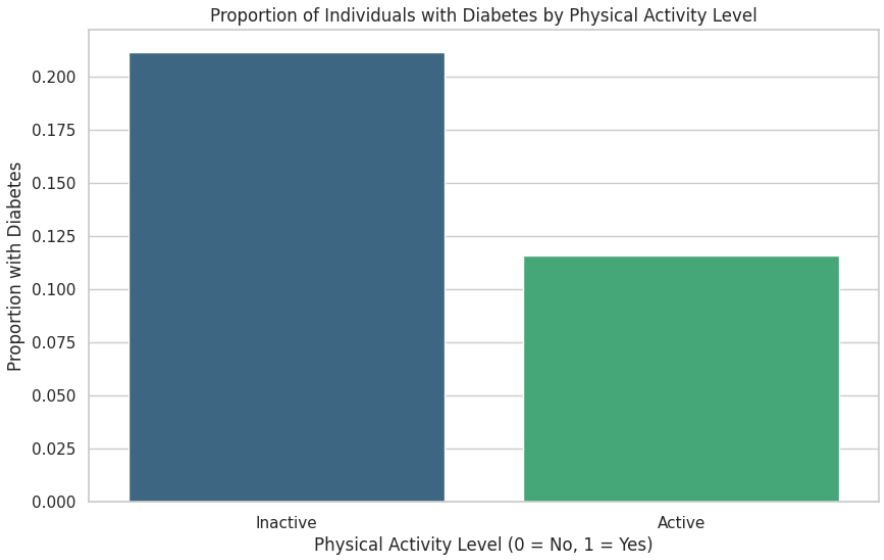


Figure 23 elucidates that a higher proportion of individuals who are inactive in physical activity have diabetes.

10. What is the impact of mental health (MentHlth) and physical health (PhysHlth) on diabetes?

Figure 24: Impact of Mental Health Days on Diabetes

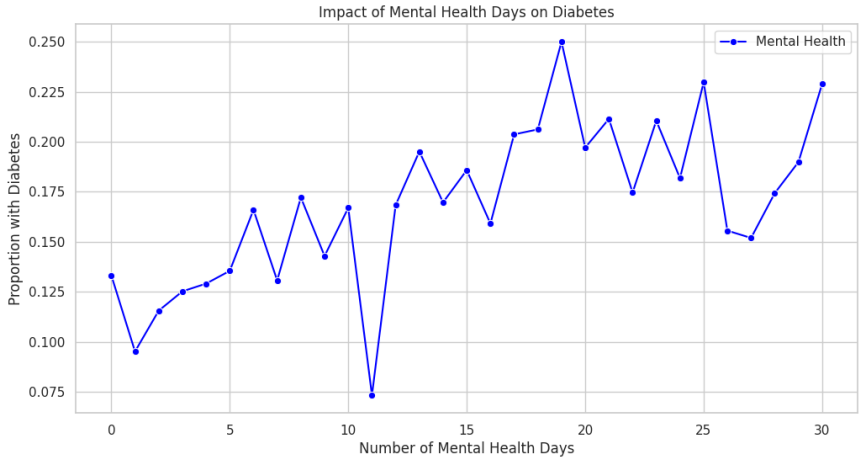
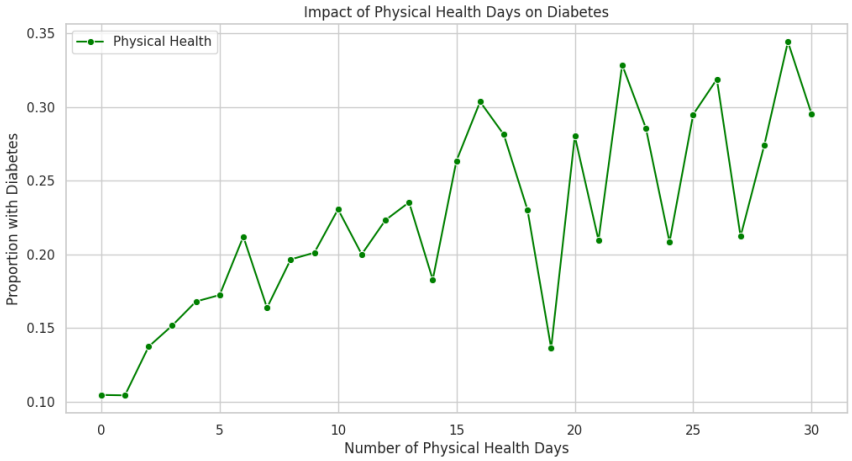


Figure 25: Impact of Physical Health Days on Diabetes



Both Figures 24 and 25 show the fluctuation across different data points but generally, they describe that poor mental and physical health are positively correlated with diabetes.

11. Is difficulty walking (DiffWalk) associated with higher diabetes incidence?

Figure 26: Difficulty Walking associated with diabetes

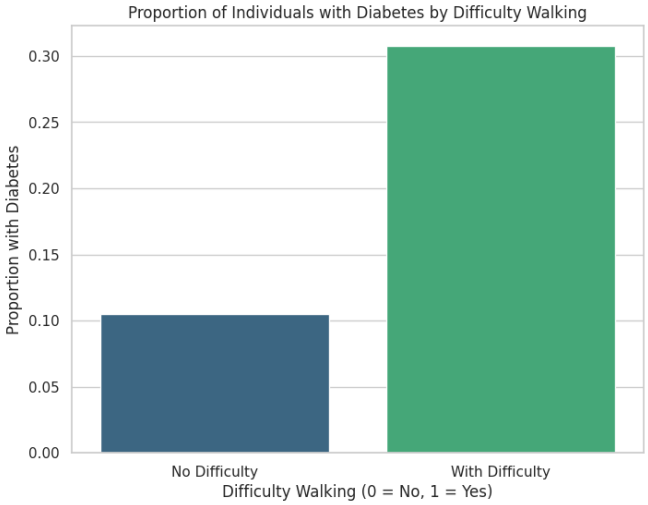


Figure 26 depicts that individuals who have difficulty in walking are more associated with diabetes.

12. Which features are most predictive of diabetes according to simple statistical tests?

Figure 27: Predictive variables for diabetes according to statistical tests

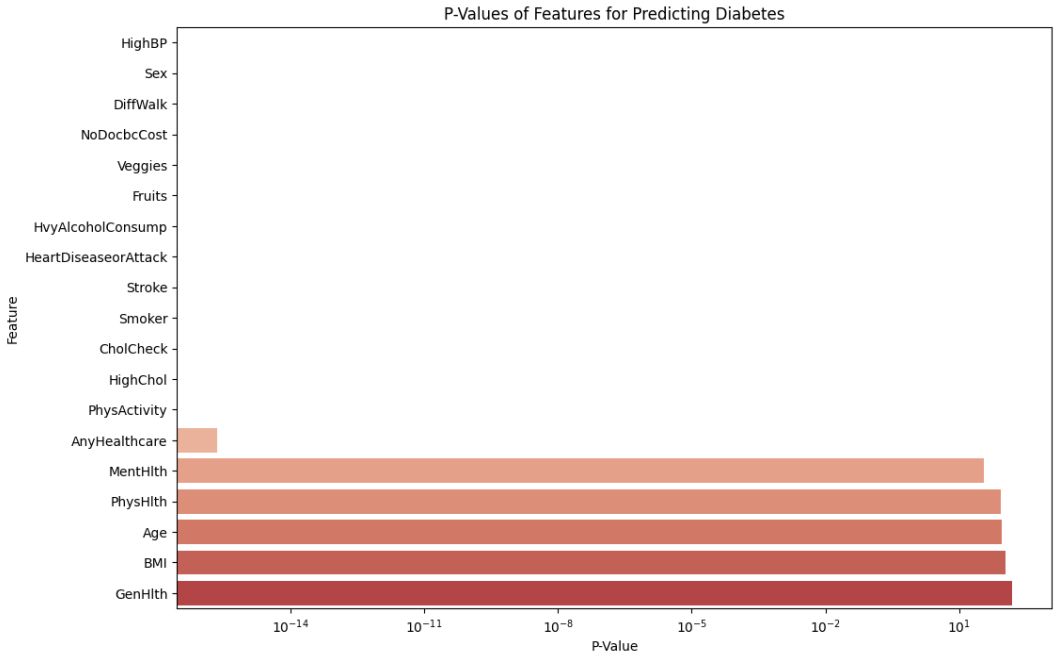


Figure 27 shows that GenHlth, BMI, Age, PhysHlth, MentHlth, AnyHealthcare have the lowest p-values which indicate that they are statistical predictors for diabetes within the dataset.

13. Finding Feature Importance for Diabetes Prediction using Classification Technique (Machine Learning)

Figure 28: Feature Importance for Diabetes Prediction

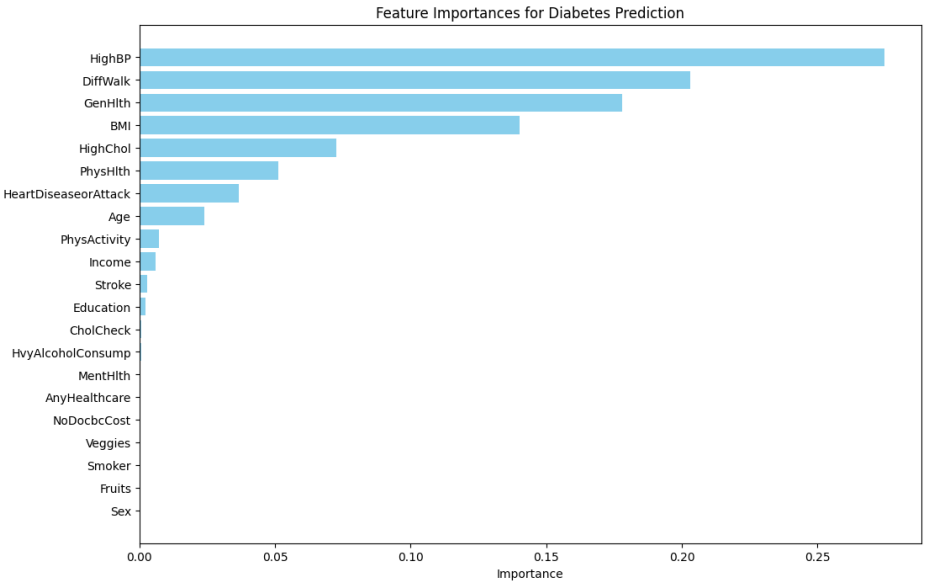


Figure 28 demonstrates that HighBP, DiffWalk, GenHlth, BMI, HighChol, and PhysHlth are critical predictors of diabetes. Using these parameters in feature selection, we can train our machine learning model to predict diabetes and test the accuracy to measure the effectiveness of this feature importance.

14. Do any variables need transformation (Normalization/Scaling) due to skewness or outliers?

Figure 29: Detecting Skewness or Outliers

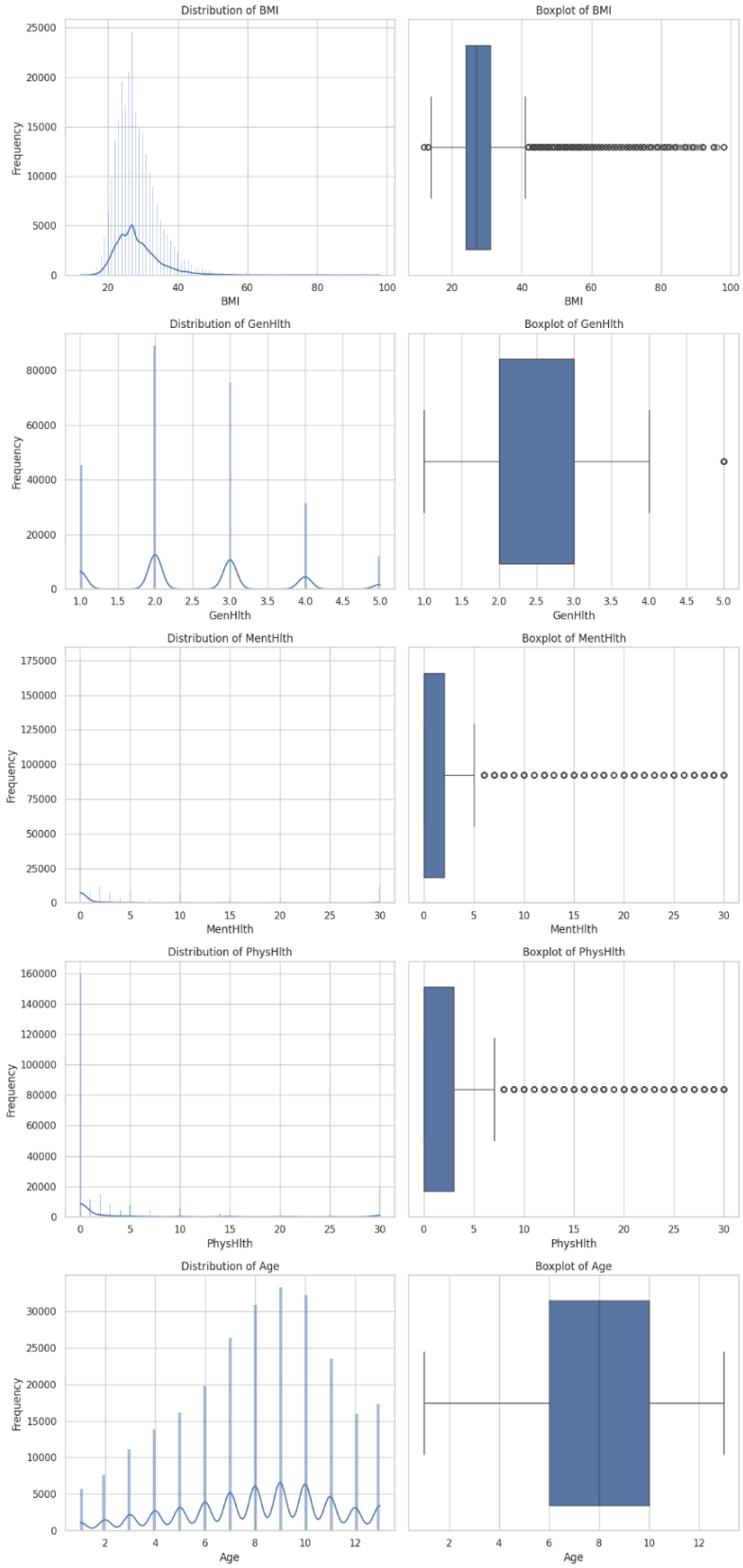


Figure 29 describes the following things about Skewness or Outliers:

1. BMI

- It is right-skewed and shows outliers on the higher end. So, it needs normalization or scaling to reduce skewness.

2. General Health (GenHlth)

- There is no need to transform but it has few outliers which is comprisable.

3. Mental Health (MentHlth) and Physical Health (PhysHlth)

- Both are showing strong right skewness with most data points showing near zero. Thus, data transformation can be done to manage the skewness and reduce outlier's effects.

4. Age

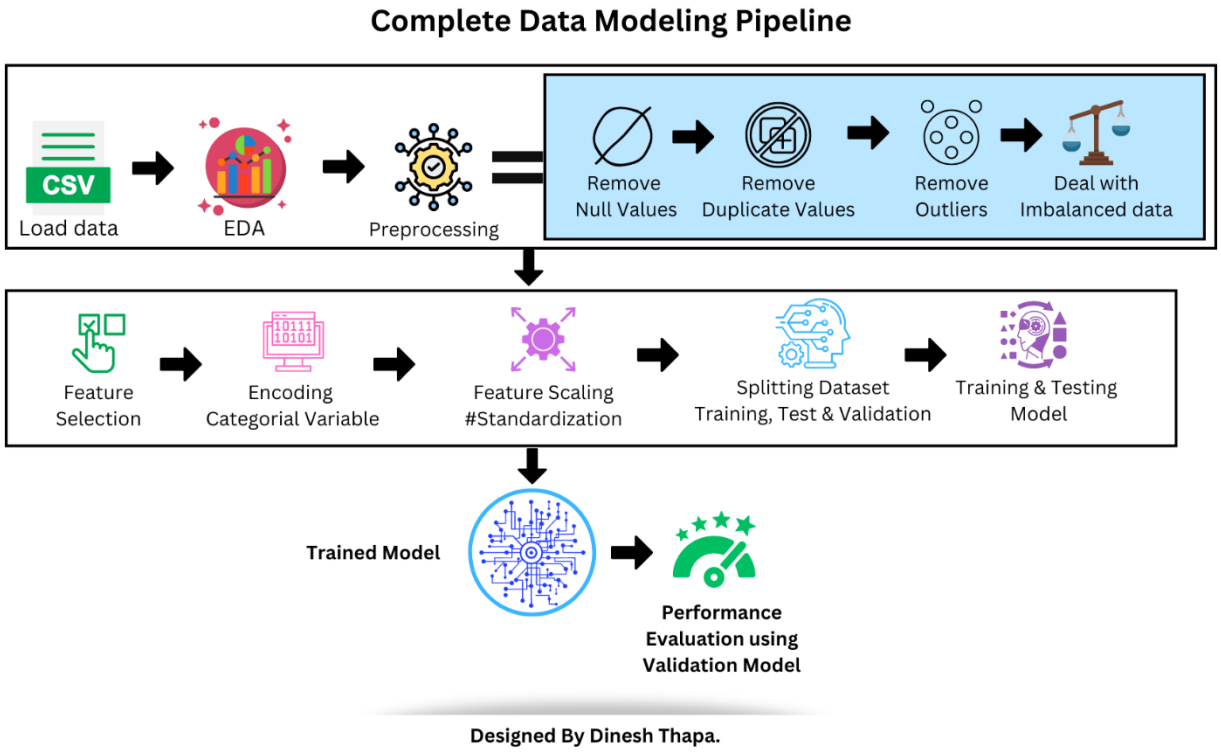
- It is represented in the dataset very well. Thus, it doesn't require any transformation.

5.2 Data Cleaning and Preprocessing

Data Cleaning and Preprocessing is the next step to be done after performing Exploratory Data Analysis. It is very crucial to ensure that data becomes usable for further analysis, and processing or can be used for machine learning modeling purposes (singh and Gaur, 2019). It is mainly performed to maintain data accuracy, completeness, consistency, timeliness, and interpretability.

Figure 30 shows the complete step-by-step process of training the machine learning model starting from loading and extracting data to cleaning, preprocessing, transformation, scaling, training, testing to model delivery.

Figure 30: Data Modeling Pipeline



The following steps are followed to perform Data Preprocessing:

1. **Remove Null Values**

Null values in a dataset can disturb our statistical analysis and machine learning model training task which require complete data for accurate predictions. So, remove null values to ensure the completeness of the data.

Code:

```
def dataCleaning(df):
    for col in df.columns:
        print(col, ":", df.filter(df[col].isNull()).count())
print("Finding NULL values:")
dataCleaning(df)
```

2. Remove Duplicates and Outliers

Duplicate and Outliers can skew the results of data analysis and predictive modeling so we have to remove them to ensure that each data record is unique and represents integrity.

In our case, BMI shows extreme outliers in datasets. Figure 29 shows that while the median BMI is below the threshold, multiple data points are unusually high. Some data points are reaching up to a BMI of 100 which is unnatural and might be the cause of data entry errors. Thus, we can remove this or standardize this using different techniques such as under-sampling, oversampling, or balancing the dataset. This will help in increasing the machine learning model's accuracy and precision score.

Code:

```
from pyspark.sql.functions import expr
from pyspark.sql.functions import when
# First, calculate the median of BMI where BMI is less than or equal to 40
median_bmi = df.filter(df['BMI'] <= 50).approxQuantile('BMI', [0.5], 0.0)[0]
# Update BMI values greater than 50 to the median BMI
df = df.withColumn('BMI', when(df['BMI'] > 50,
median_bmi).otherwise(df['BMI']))
```

3. Feature Selection

In machine learning algorithms, Feature Selection is about identifying and choosing the most relevant variables for predicting a specific outcome. From figures 27 and 28, it is clear that features like High Blood Pressure (HighBP), General Health (GenHlth), Body Mass Index (BMI), Physical Health (PhysHlth), Age, High

Cholesterol (HighChol), difficulty walking (DiffWalk), Heart Disease or Attack, and Smoking are identified as important predictors for diabetes. These selections are based on exploratory data analysis (EDA) performed.

Selected Features = (HighBP, HighChol, BMI, Smoker, HeartDiseaseorAttack, GenHlth, PhysHlth, DiffWalk, Age)

4. Encoding Categorical Variables

In our dataset, multiple categorical variables need to be converted into numerical features before applying machine learning models because they only work with numbers, not strings.

Code:

```
# Encoding categorical columns and assembling a feature vector

categoricalColumns = ['Income','Education']
```

5. Splitting dataset into training and test set

In this step, we split our diabetes dataset into three subsets as follows: 60% for the training set, 20% for testing, and 20% for validation. As the quantity of data increases regularly, the proportion of testing and validation data decreases.

Code:

```
# Split the data into training, test, and validation sets

train_data, test_data, val_data = df_transformed.randomSplit([0.6, 0.2, 0.2],
seed=42)
```

6. Feature Scaling

Feature Scaling is used to make variables in the same range. There are different types of feature scaling techniques such as Normalization, Standardization and Robust Scaling.

In our project, we have implemented Standardization in classification technique as follows:

Code:

```
#feature_scaling

from pyspark.ml.feature import StandardScaler

# Scale features using StandardScaler

scaler = StandardScaler(inputCol="features", outputCol="scaledFeatures",
withStd=True, withMean=True)

stages += [scaler]
```

7. Dealing with Imbalanced Data

Figures 13 and 14 show the imbalanced data in our diabetes dataset. For instance, 13.9% of data records are diabetic and 86.9% are non-diabetic. Due to imbalanced data, the machine learning model will not perform well which will lead to failure in predicting diabetic cases accurately. To solve this issue, we can either oversample the minority class or sample the majority class. In our case, we perform under-sampling which means filtering or decreasing data from the majority class to match with the minority class.

Code:

```
diabetic_data = df.filter(df['Diabetes_binary'] == 1).limit(35346)

non_diabetic_data = df.filter(df['Diabetes_binary'] == 0).limit(35346)

# Concatenate the two DataFrames

df = diabetic_data.union(non_diabetic_data)
```

Step 7 will only be applicable in case of implementing classification-related problems or using classification techniques such as Logistic Regression.

6 Machine Learning: Classification and Clustering

Machine Learning is a type of statistical algorithm that can learn from the patterns and behaviors of the dataset and perform specific tasks guided by that pattern-based instruction. It is applicable in multiple fields including bioinformatics, medical fields, e-commerce recommendation engines, and neural networks (Baştanlar and Özuysal, 2014).

Mainly, there are three types of machine learning algorithms including supervised, unsupervised, and reinforcement learning (Zhang, 2010). Among them, both supervised and unsupervised algorithm will be implemented in this project of predicting diabetes.

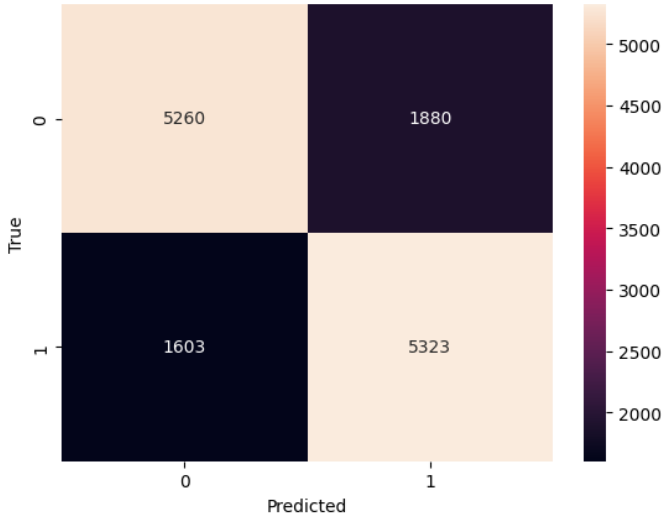
7 Classification Techniques

Classification Technique is a type of supervised machine learning algorithm that uses labelled data as an input to train the model and can predict the outcome of the subsequent data based on the patterns of its training data (Richards, 2022). So, the probability of predicting the outcome depends on the patterns and behaviors of the training data. There are various types of classification techniques such as Logistic Regression, Random Forest, and Gradient Boosted Trees which will be demonstrated in this project to predict the diabetes.

7.1 Logistic Regression

Logistic Regression is a type of classification technique which predicts the outcome of an input variable based on the statistical formula used in the logistic function. In this project, we apply Logistic Regression technique to predict the diabetes on the basis of different predictors based on the training dataset applied to the model and it gives result in binary such as presence or absence of disease (Nick and Campbell, 2007).

Figure 31: Logistic Regression - Confusion Matrix



Logistic Regression Test Accuracy (AUC): 0.8289606472100819

Logistic Regression Validation Accuracy (AUC): 0.8323600647693522

These AUC scores indicates that the logistic regression model has a very good capability of identifying both diabetic and non-diabetic cases. The figure 30 shows the confusion matrix for logistic regression which shows the following key information: -

- True Negative (5260) means that the model correctly predicted the non-diabetic cases.
- False Positive (1880) means that the model incorrectly predicted these cases as diabetic when they were not.
- False Negative (1603) means that the model incorrectly predicted these cases as non-diabetic when they were actually diabetic.
- True Positive (5323) means that the model correctly predicted the diabetic cases.

The consistency between Test accuracy and Validation accuracy ensures that Logistic Regression model performs well in predicting diabetic cases.

Model Evaluation:

Figure 32: Model Evaluation of Logistic Regression

```
Precision for class '1' (having diabetes):  0.7389976398722754
Recall for class '1' (having diabetes):  0.7685532775050534
F1-score for class '1' (having diabetes):  0.7534857385519144
Support for class '1' (having diabetes):  6926
Precision for class '0' (not having diabetes):  0.7664286755063383
Recall for class '0' (not having diabetes):  0.7366946778711485
F1-score for class '0' (not having diabetes):  0.7512675855173891
Support for class '0' (not having diabetes):  7140
```

Figure 31 shows Precision, Recall, F1-score, and Support score for the logistic regression model. For class '1' (having diabetes), the F1-score is 75.35% which means a balanced measure of both precision and recall indicates good model accuracy.

7.2 Random Forest Classifier

Random forest classifier is an ensemble machine learning technique takes three hyperparameters before training the model such as node size, the number of trees, and the number of sample features (Speiser *et al.*, 2019). It is implemented in this project to predict the diabetic or non-diabetic condition based on the list of different features.

Figure 33: Random Forest Classifier - Confusion Matrix

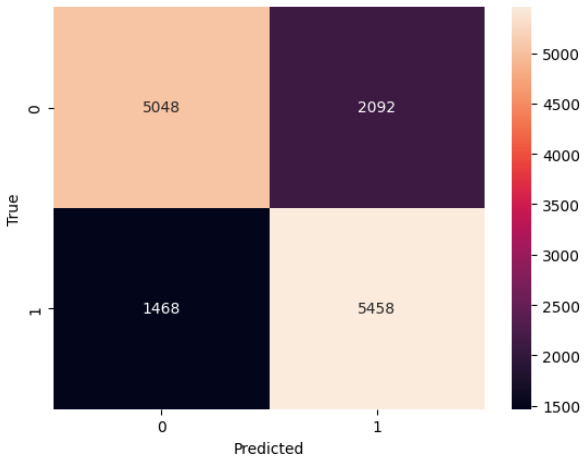


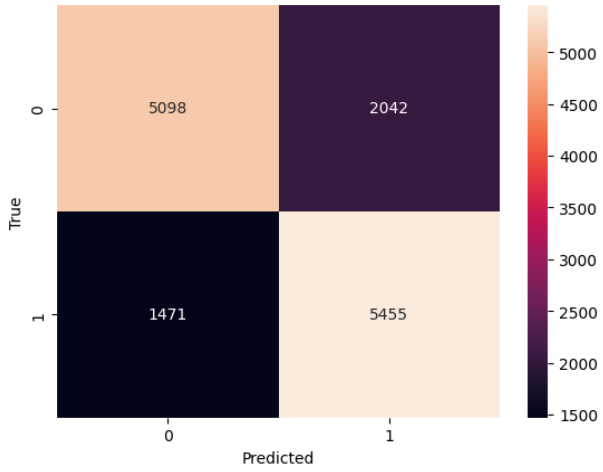
Figure 32 shows the confusion matrix which is generated after running Random Forest Classifier algorithm. The accuracy and model performance are almost same in both logistic regression and random forest algorithm.

The accuracy score (AUC) of random forest algorithm is 0.8202784073490791.

7.3 Gradient Boosted Trees Classifier

Gradient Boosted Tree (GBT) is a type of machine learning technique where multiple models are trained sequentially to correct the errors made by previous ones (Saberian *et al.*, 2019). This algorithm is implemented to predict diabetes using the same training and test dataset to compare the results between different machine learning algorithm.

Figure 34: GBT - Confusion Matrix



The accuracy of Gradient Boosted Trees (AUC) is 0.8267305593909525

Figure 33 also clarifies that GBT algorithm is also providing similar accuracy ratio just like Logistic Regression and Random Forest Classifier.

7.4 Model Scalability and Runtime Evaluation

Figure 35: Model Scalability of three different algorithms

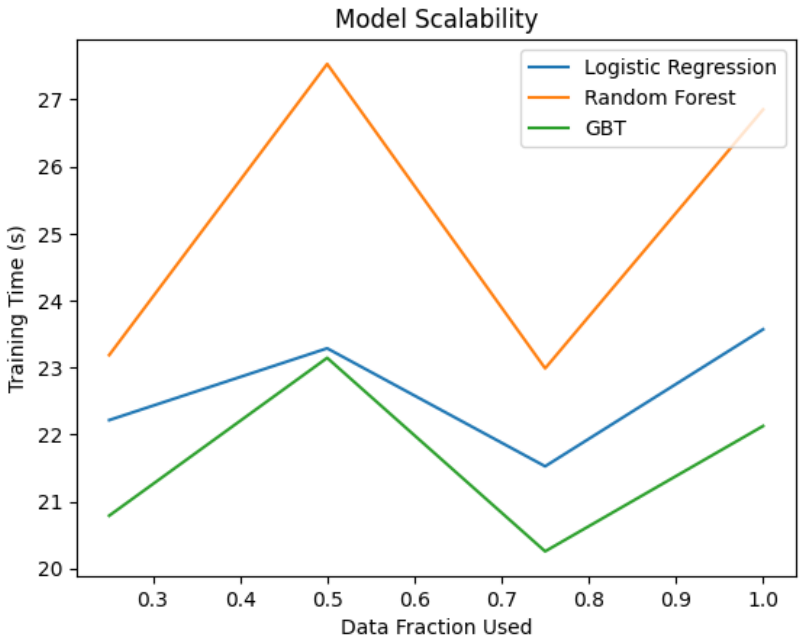


Figure 34 depicts the model scalability of three different algorithms: Logistic Regression, Random Forest, and Gradient Boosted Trees.

From the graph, it is clear that Logistic Regression is consistently the fastest in terms of training time which makes it a good machine learning algorithm for large datasets. GBT shows variability in training time which can be varied based on the sizes of its training data and multiple parameters. Random Forest shows the highest training times thus, it is not suitable for very large datasets or real-time applications.

The model runtime evaluation is as follows: -

Logistic Regression training and prediction time: 34.01 seconds

Random Forest training and prediction time: 25.56 seconds

Gradient-Boosted Trees training and prediction time: 25.72 seconds

8 Clustering Techniques

The clustering technique is a type of unsupervised machine learning method that is used to group the data points based on their similarities. It is the process of finding similar structures of behaviors in an uncategorized dataset (Sisodia and Singh, n.d.). There are different types of clustering algorithms such as K-means clustering and the Gaussian Mixture Model which will be implemented in our project on diabetes prediction.

8.1 K-means clustering

K-means clustering is the most popular clustering algorithm to detect similar types of groups or clusters within a dataset (Sinaga and Yang, 2020). In the diabetes project, we use k-means clustering to find different clusters having the same type of behaviors so that we can recommend customized healthcare plans or diagnostic plans to individuals.

Figure 35 shows the silhouette scores for different numbers of clusters (k) used in the k-means clustering of a diabetes dataset. The silhouette score measures how similar a node is to its cluster compared to other clusters and a high silhouette score means better defined clusters. The highest score in the figure is at k=9, which tells that nine clusters provide the best separation of data points.

Figure 36: Silhouette Scores for Different Numbers of Clusters

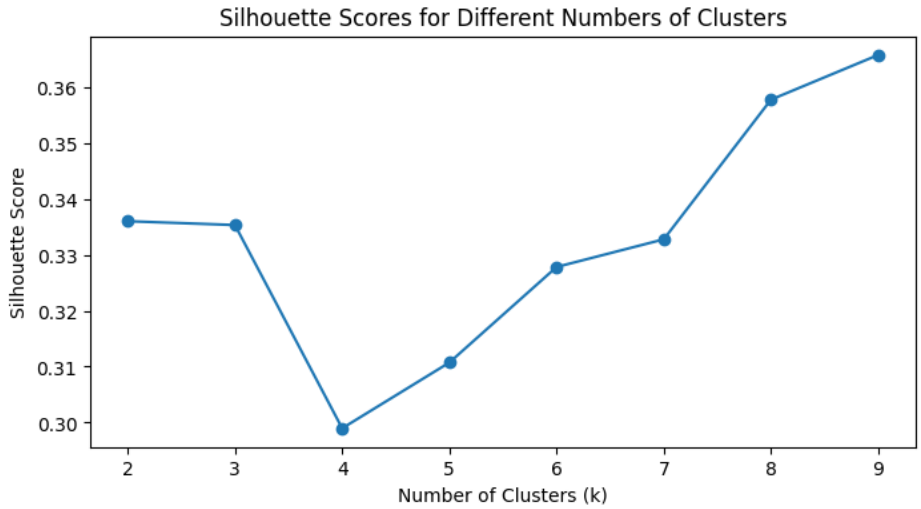


Figure 37: K-Means Clustering | PCA of Clusters

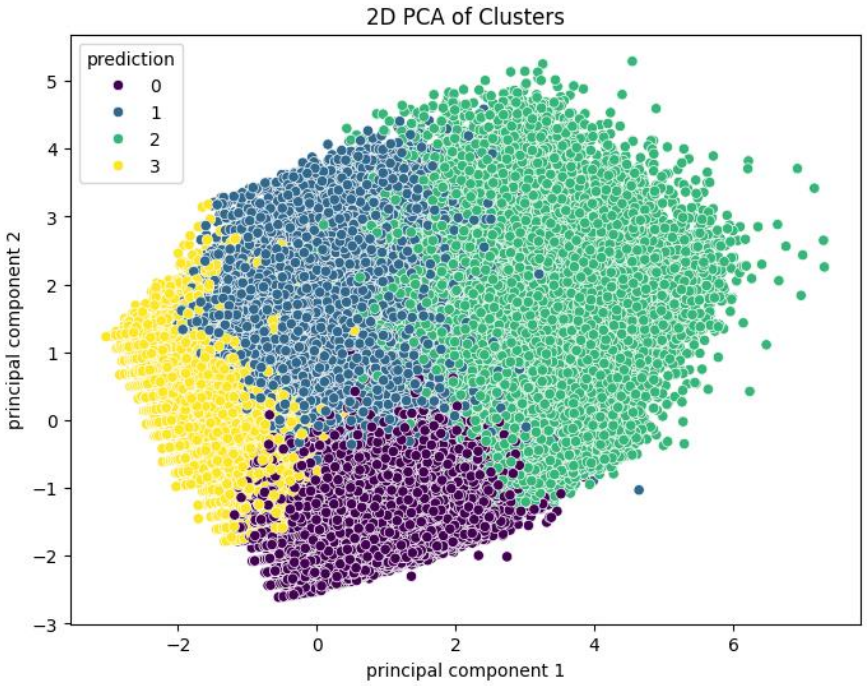
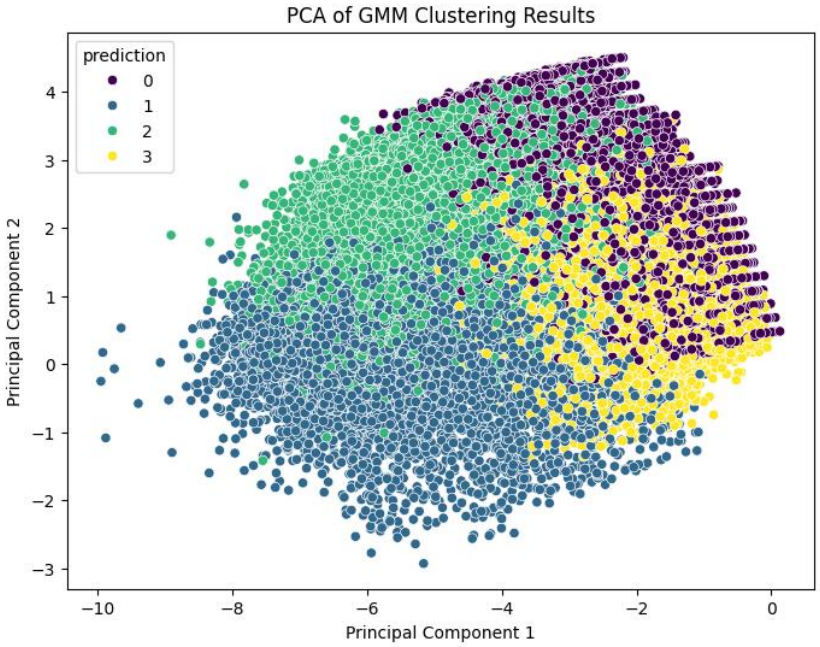


Figure 36 shows that k-means clustering has successfully identified the groups of clusters with common characteristics within the diabetes dataset. Based on different clusters having different behaviors of diabetic and non-diabetic patterns, the algorithm will be able to recommend diabetes-related advice and consultation.

8.2 Gaussian Mixture Model (GMM)

Gaussian Mixture Model is a type of probabilistic model used for clustering datasets into multiple groups based on the mixture of several gaussian distributions (Yu and Deng, 2015).

Figure 38: PCA of GMM Clustering Results



Cluster sizes: [31322, 7470, 20341, 11559]

Log Likelihood: -394006.3468172617

Figure 39: Cluster Statistics from GMM Model

```
Cluster 0 center (mean): [1.0234726469903248,0.94555895684809,4.0769009534921645,3.054226996799185,2.094275808846385,0.00014656446748367448,0.00010330031083101196,2.8141957538881375]
Cluster 1 center (mean): [0.8855425959595795,1.161352967943505,4.47070995046104,2.8224313324734824,3.3527640083678008,1.9781426819118941,2.630219743590371,2.1033344208395603]
Cluster 2 center (mean): [2.013523078799383,1.3717736091457828,4.583318367364164,3.2266469570949203,3.027604954683877,1.1058640476683612,0.4314181566039409,2.354678749739346]
Cluster 3 center (mean): [0.00040393106584723495,0.6975780508121225,3.9298554856232335,2.415983602371368,2.1718326195645186,0.2926637293315908,0.35781547999689156,2.863761797432341]
+----------+---------------+
|prediction|avg(prediction)|
+----------+---------------+
|         1|            1.0|
|         3|            3.0|
|         2|            2.0|
|         0|            0.0|
+----------+---------------+
```

Figure 37 shows the visualization of GMM clustering using PCA through reducing dimensionality. It has identified four clusters: 31322, 7470, 20341, and 11559. Thus, based on different diabetes-related features and attributes, it is represented into multiple clusters showing different groups from the used dataset. Based on this cluster, we can identify potential patients, and non-patients and provide them with customized feedback and recommendation systems powered by machine learning.

9 Graph Analysis

A Graph Database is a non-relational database that uses graph structure for storing, managing, and processing data in the forms of nodes, edges, properties, and their relationships (Guia *et al.*, 2017). There are different graph databases available such as Neo4j, Arango DB, Neptune, Graph Base, and many more but we are going to implement Neo4j for our project of analyzing the "Catch the Pink Flamingo" dataset.

Graph Analysis involves discovering and finding different types of hidden relationships between its nodes, edges, and properties to find meaningful trends, patterns, and insights using the graph data and special graph database management system and its query language. Graph Analytics is very useful for businesses as they can drive real business value and make data-driven decisions based on their complete analysis report. Furthermore, using graph analytics, organizations could understand complex volumes of data by using different graph algorithms and visualization techniques.

9.1 Dataset Description

For graph analysis, we are going to use the "Catch the Pink Flamingo" graph database for chats. This dataset consists of 6 files which are shown below with their column names.

Schema of the Chat Data from Catch the Pink Flamingo

Table 3: Schema of Chat Data

File Name	Column Names
chat_create_team_chat.csv	userid, teamid, TeamChatSessionID, timestamp
chat_item_team_chat.csv	userid, teamchatsessionid, chatitemid, timestamp
chat_join_team_chat.csv	userid, TeamChatSessionID, timestamp
chat_leave_team_chat.csv	userid, teamchatsessionid, timestamp
chat_mention_team_chat.csv	ChatItem, userid, timestamp
chat_respond_team_chat.csv	chatid1, chatid2, timestamp

Note: These files are hosted on my GitHub profile which is used below to load the dataset directly from the GitHub cloud.

9.2 Loading CSV files into Neo4J for Graph Analysis

A. Loading chat_create_team_chat.csv file

```
LOAD CSV FROM
'https://raw.githubusercontent.com/DineshThapaX/neo4j_sample_project/main/chat
_create_team_chat.csv' AS row

WITH row
```

```
WHERE row[0] IS NOT NULL AND row[2] IS NOT NULL

MERGE (u:User {id: row[0]})

MERGE (t:TeamChatSession {id: row[2]})

MERGE (u)-[:CREATED {timestamp: toInteger(row[3])}]->(t);
```

B. Loading chat_item_team_chat.csv file

```
LOAD CSV FROM
'https://raw.githubusercontent.com/DineshThapaX/neo4j_sample_project/main/chat
_item_team_chat.csv' AS row

FIELDTERMINATOR ','  // As your values seem comma-separated

WITH row

WHERE row[0] IS NOT NULL AND row[1] IS NOT NULL AND row[2] IS NOT NULL

MERGE (u:User {id: row[0]})

MERGE (t:TeamChatSession {id: row[1]})

MERGE (c:ChatItem {id: row[2]})

MERGE (u)-[:CREATED_CHAT_ITEM {timestamp: toInteger(row[3])}]->(c)

MERGE (c)-[:PART_OF {timestamp: toInteger(row[3])}]->(t);
```

C. Loading chat_join_team_chat.csv file

```
LOAD CSV FROM
'https://raw.githubusercontent.com/DineshThapaX/neo4j_sample_project/main/chat
_join_team_chat.csv' AS row

FIELDTERMINATOR ','  // Assuming the delimiter is a comma

WITH row

WHERE row[0] IS NOT NULL AND row[1] IS NOT NULL

MERGE (u:User {id: row[0]})

MERGE (t:TeamChatSession {id: row[1]})

MERGE (u)-[:JOINS {timestamp: toInteger(row[2])}]->(t);
```

D. Loading chat_leave_team_chat.csv file

```
LOAD CSV FROM
'https://raw.githubusercontent.com/DineshThapaX/neo4j_sample_project/main/chat
_leave_team_chat.csv' AS row

FIELDTERMINATOR ','  // Adjust if your delimiter is different
```

```
WITH row

WHERE row[0] IS NOT NULL AND row[1] IS NOT NULL

MERGE (u:User {id: row[0]})

MERGE (t:TeamChatSession {id: row[1]})

MERGE (u)-[:LEAVES {timestamp: toFloat(row[2])}]->(t);
```

E. Loading chat_mention_team_chat.csv file

```
LOAD CSV FROM
'https://raw.githubusercontent.com/DineshThapaX/neo4j_sample_project/main/chat
_mention_team_chat.csv' AS row

WITH row

WHERE row[0] IS NOT NULL AND row[1] IS NOT NULL

MERGE (c:ChatItem {id: row[0]})

MERGE (u:User {id: row[1]})

MERGE (c)-[:MENTIONED {timestamp: toInteger(row[2])}]->(u);
```

F. Loading chat_respond_team_chat.csv file

```
LOAD CSV FROM
'https://raw.githubusercontent.com/DineshThapaX/neo4j_sample_project/main/chat
_respond_team_chat.csv' AS row

WITH row

WHERE row[0] IS NOT NULL AND row[1] IS NOT NULL

MERGE (c1:ChatItem {id: row[0]})

MERGE (c2:ChatItem {id: row[1]})

MERGE (c1)-[:RESPONDS_TO {timestamp: toInteger(row[2])}]->(c2);
```

Now, using all of these queries, all the csv files are loaded into Neo4j database successfully.

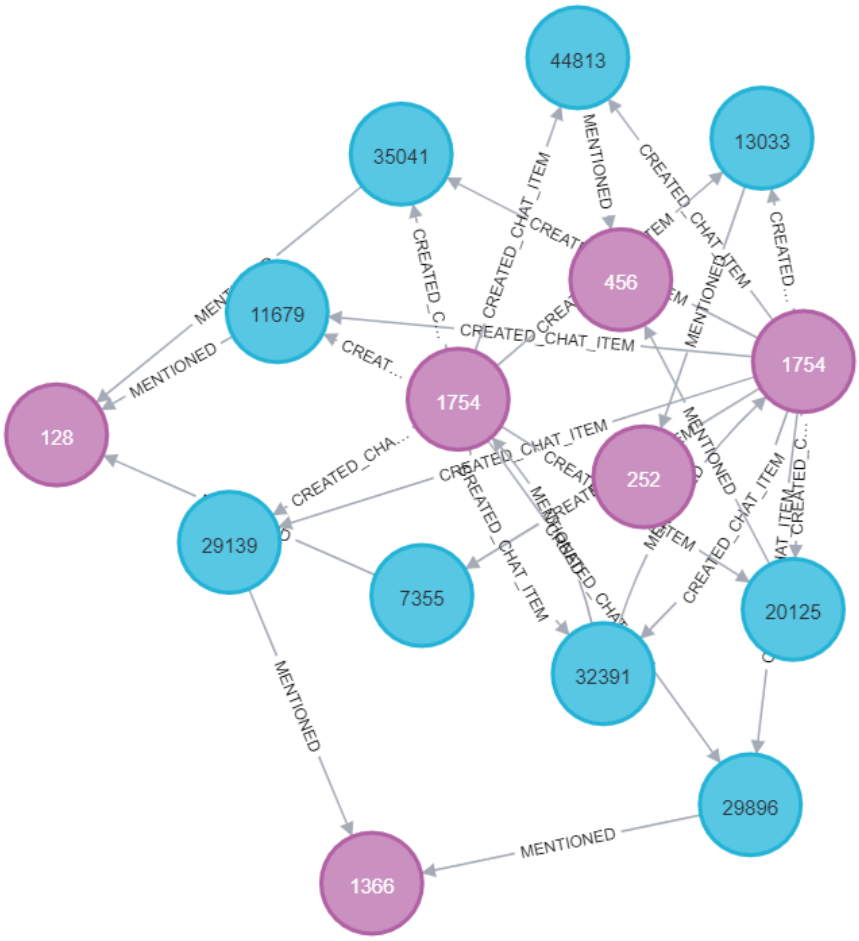
9.3 Graph Analysis and Visualization using Cypher Query

Here, we perform different combination of analytical queries and visualize them using graphs and relevant figures.

1. **Who are the key influencers based on Chat Item Creation and Mentions?**

The figure 39 shows the top 10 key users based on the number of chat items they have created and based on their mentions in the game. The query returns each influential user along with chat items they created and user they mentioned.

Figure 40: Key Influencers in the Chat



Cypher Query:

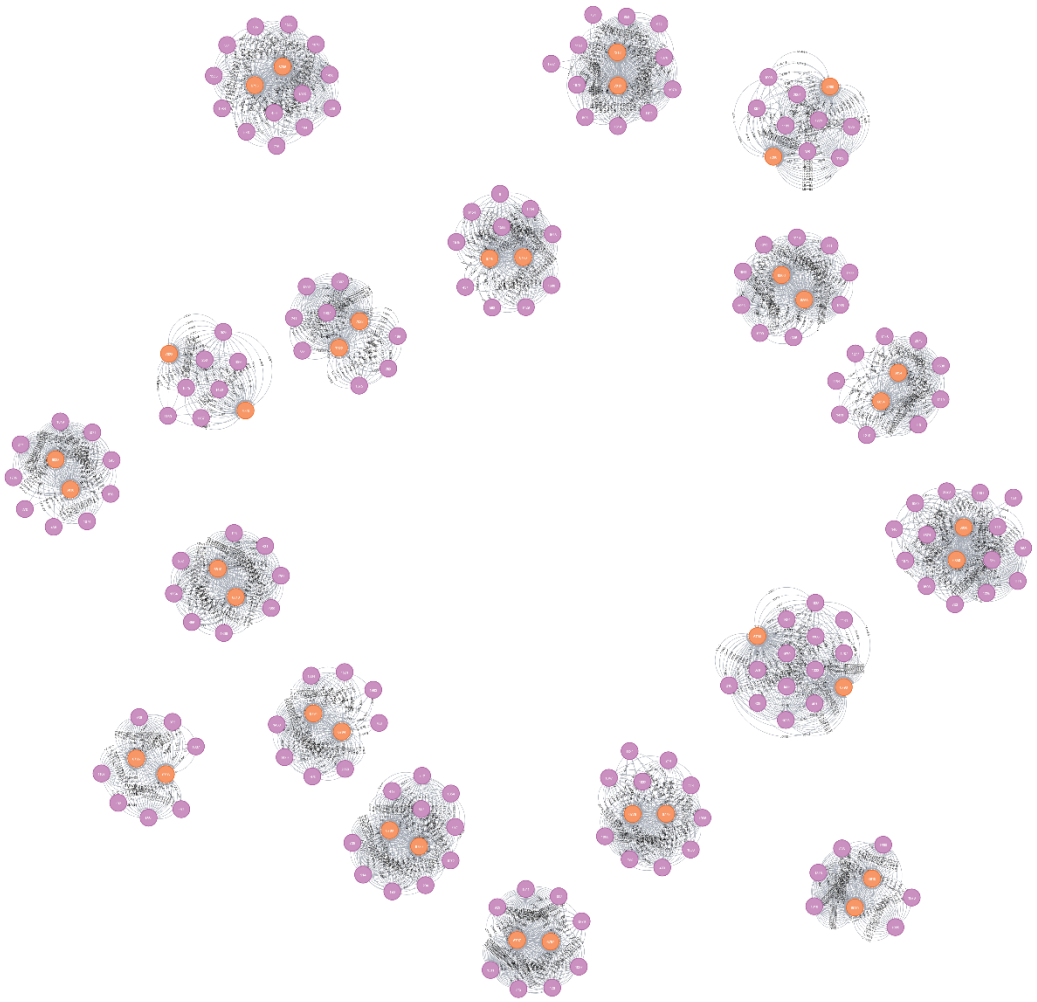
```
// Match users and the chat items they created
```

```
MATCH (u:User)-[createdRel:CREATED_CHAT_ITEM]->(c:ChatItem)

// Count the items created by each user

WITH u, c, count(c) AS ItemsCreated

// Match the mentions from these chat items to other users

OPTIONAL MATCH (c)-[mentionedRel:MENTIONED]->(other:User)

// Return results including graph elements

WITH u, ItemsCreated, c, other, count(other) AS TimesMentioned

ORDER BY TimesMentioned DESC, ItemsCreated DESC

LIMIT 10

RETURN u, collect(c) AS ChatItems, collect(other) AS MentionedUsers,
ItemsCreated, TimesMentioned;
```

2. **How chat session is interconnected through user participation?**

Figure 40 shows the chat sessions based on user involvement in different chat sessions. It shows the top 25 pairs with the highest number of user participation.

Figure 41: Chat Session Interconnection through user participation



Cypher Query:

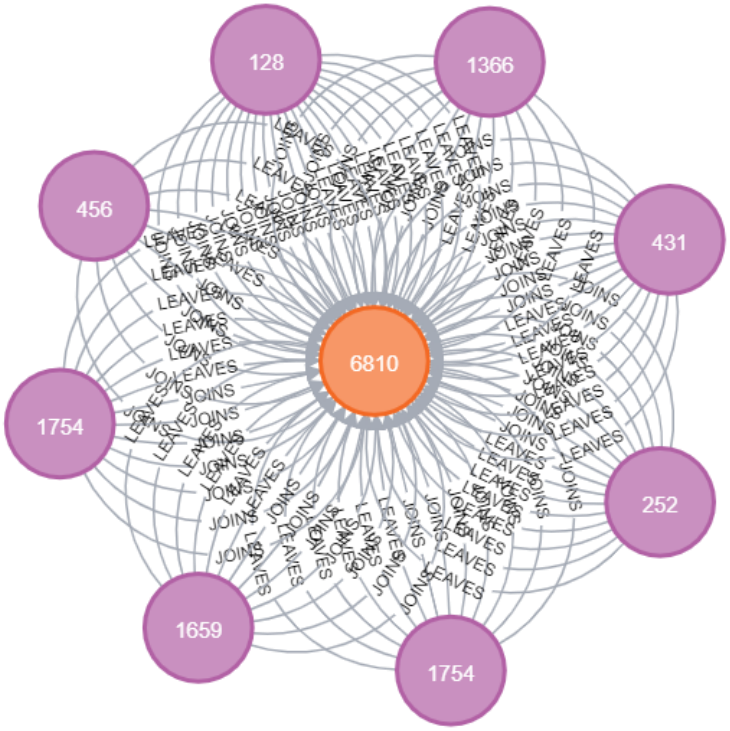
```
// Match users who join multiple sessions

MATCH (s1:TeamChatSession)<-[j1:JOINS]-(u:User)-[j2:JOINS]-
>(s2:TeamChatSession)

WHERE id(s1) < id(s2)

WITH s1, s2, u, COUNT(u) AS SharedUsers

ORDER BY SharedUsers DESC, id(s1), id(s2)

LIMIT 25


// Match other users joining these sessions for broader context
```

```
OPTIONAL MATCH (s1)<-[j1Other:JOINS]-(uOther:User)

OPTIONAL MATCH (s2)<-[j2Other:JOINS]-(uOther)

WITH s1, s2, COLLECT(DISTINCT u) AS ConnectingUsers, COLLECT(DISTINCT uOther)
AS OtherUsersInSessions


// Return all sessions, users, and relationships for visualization

RETURN s1, s2, ConnectingUsers, OtherUsersInSessions,

       size(ConnectingUsers) AS NumberOfConnectingUsers,

       size(OtherUsersInSessions) AS TotalUsersInSessions;
```

3. Visualize the User Interaction Networks and User Interactions through Chat Items

Figure 41 shows the visualization of a user interaction network in which multiple users interact in a chat during their game in a team chat session. It shows only 100 interactions in the graph.

Figure 42: User Interaction Network

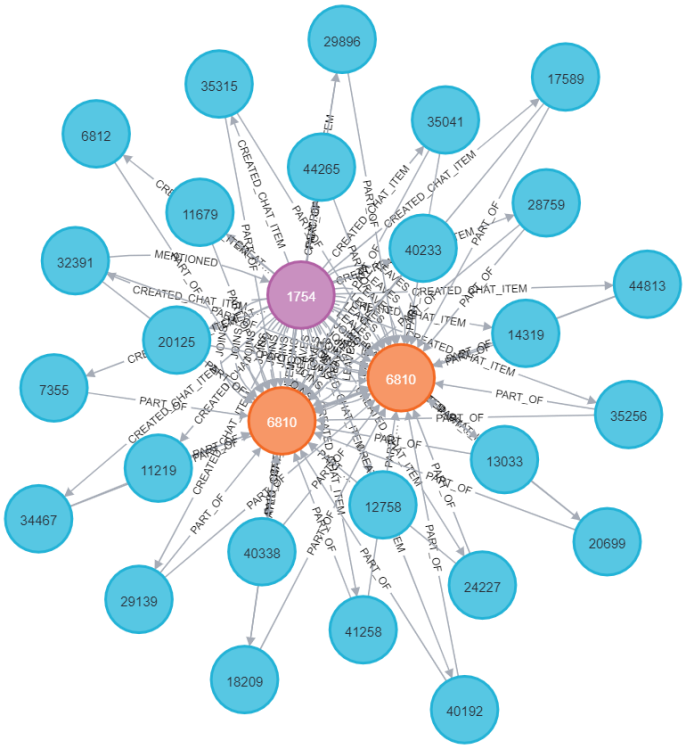


Cypher Query for User Interaction Networks:

```
MATCH (u:User)-[:JOINS]->(t:TeamChatSession)<-[:JOINS]-(u2:User)
RETURN u, t, u2
LIMIT 100;
```

Figure 42 shows the user interactions through chat items. It only highlights 50 user interactions within chat sessions.

Figure 43:User Interactions through Chat Items



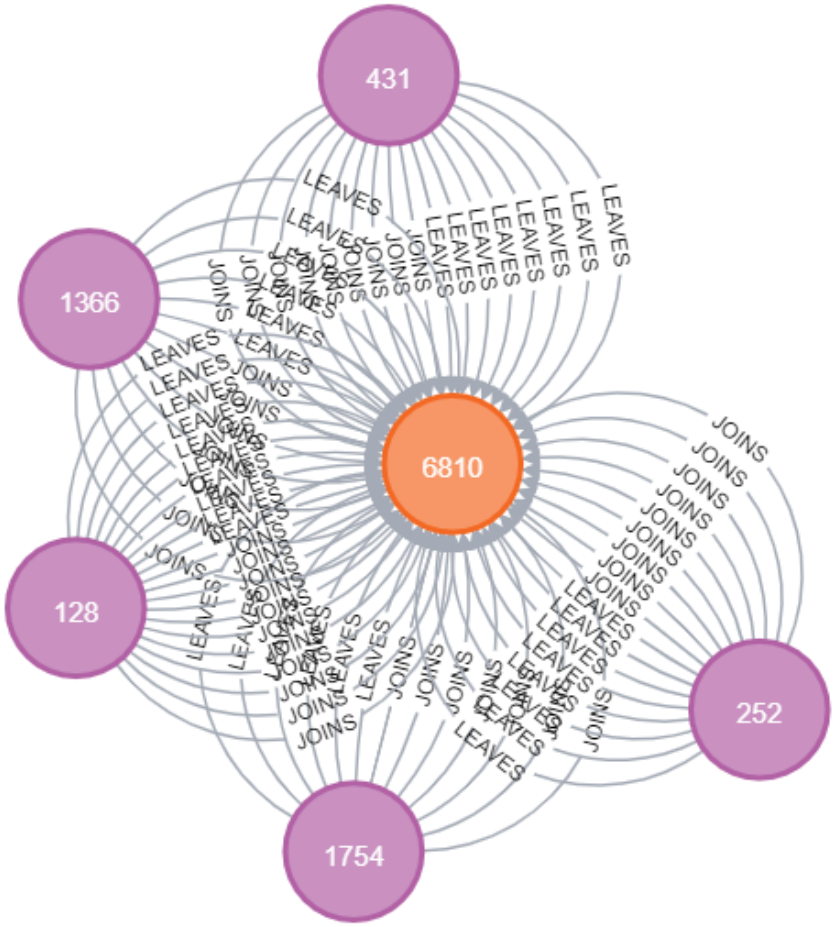
Cypher Query for user Interactions through Chat Items:

```
MATCH (u:User)-[r:CREATED_CHAT_ITEM]->(c:ChatItem)-[:PART_OF]-
>(t:TeamChatSession)
RETURN u, r, c, t
LIMIT 50;
```

4. Visualize Community Interaction by Joining Sessions

Figure 43 illustrates the community interactions in the context of joining chat sessions. The results are limited to 100 instances to provide a clear graph.

Figure 44: Showing Community Interaction by Joining Sessions



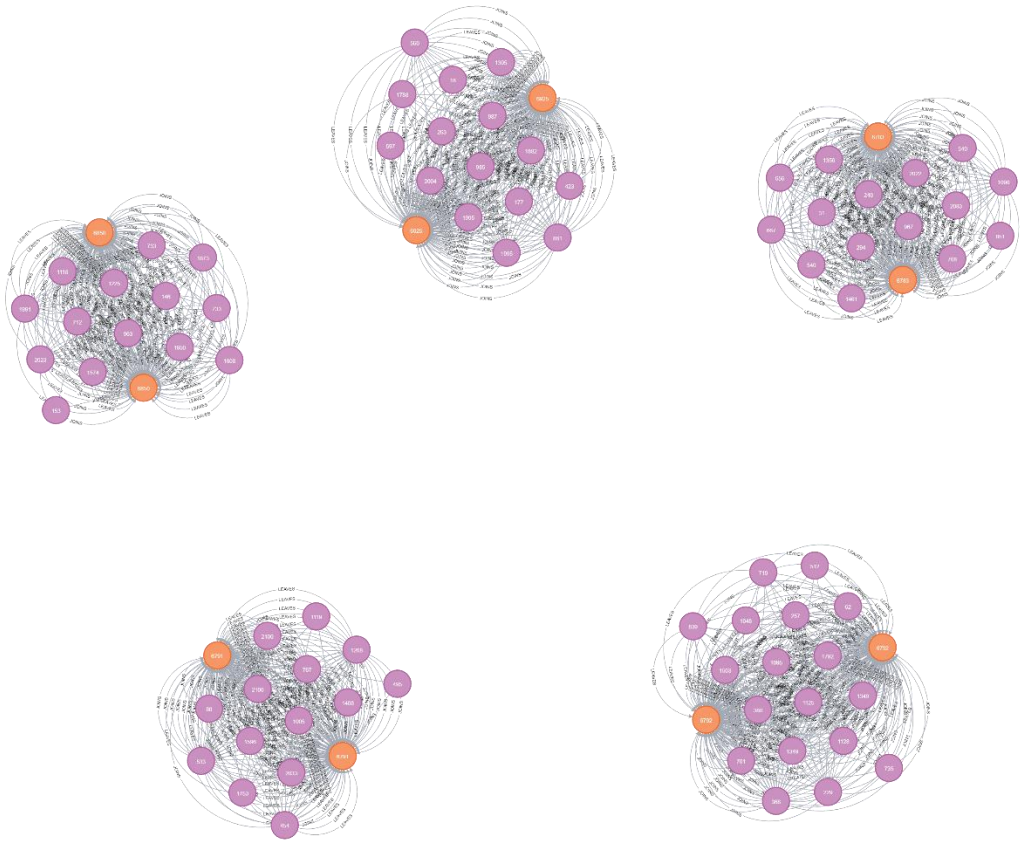
Cypher Query:

```
MATCH (u1:User)-[:JOINS]->(t:TeamChatSession)<-[:JOINS]-(u2:User)
WHERE id(u1) < id(u2)
RETURN u1, u2, t
LIMIT 100;
```

5. Visualize the evolution of chat sessions over time in a game

Figure 44 displays the evolution of different chat sessions within a game over a period of different timestamps.

Figure 45: Evolution of Chat Sessions



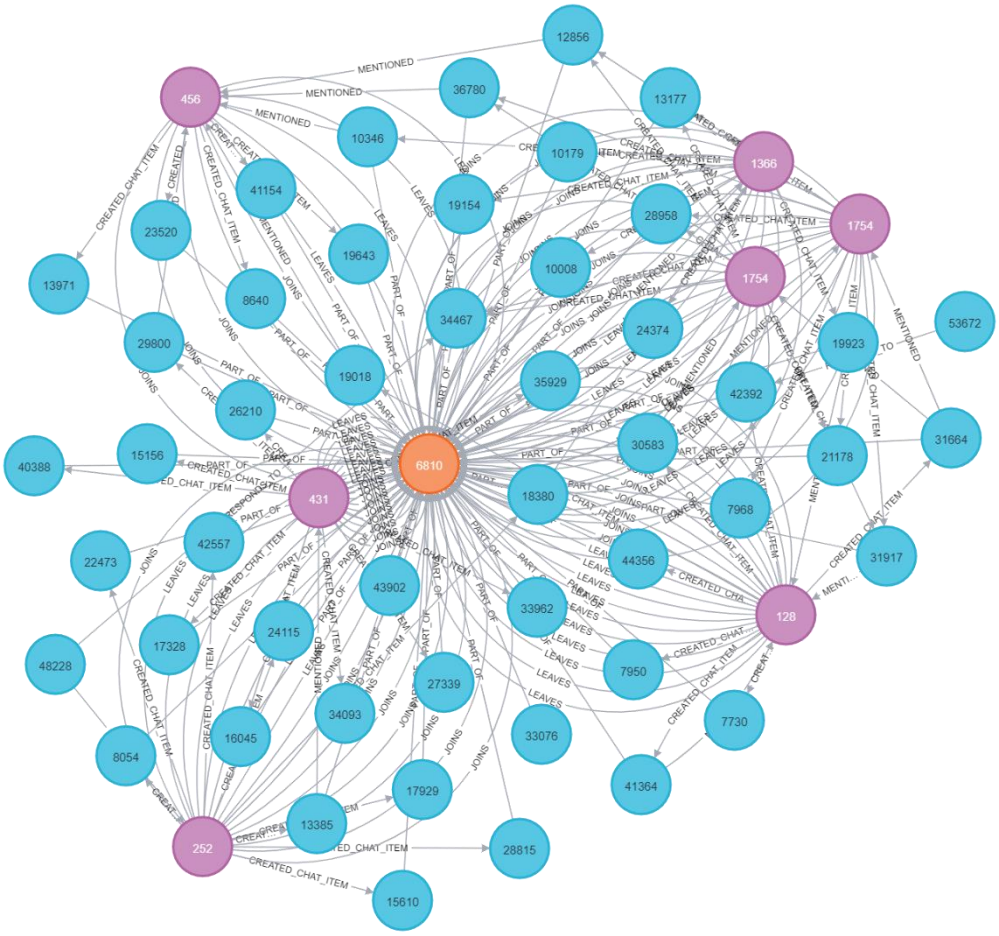
Cypher Query:

```
MATCH (t:TeamChatSession)
OPTIONAL MATCH (t)<-[:JOINS]-(u:User)
OPTIONAL MATCH (t)<-[:CREATES]-(ci:ChatItem)
RETURN t, collect(u) AS participants, collect(ci) AS chatItems
ORDER BY size(participants) DESC
LIMIT 10;
```

6. Visualize the lifecycle of a chat session created in a game.

Figure 45 shows the lifecycle of a complete chat session initiated in a game. It shows how community interactions occur within a chat session.

Figure 46: Lifecycle of a Chat Session



Cypher Query:

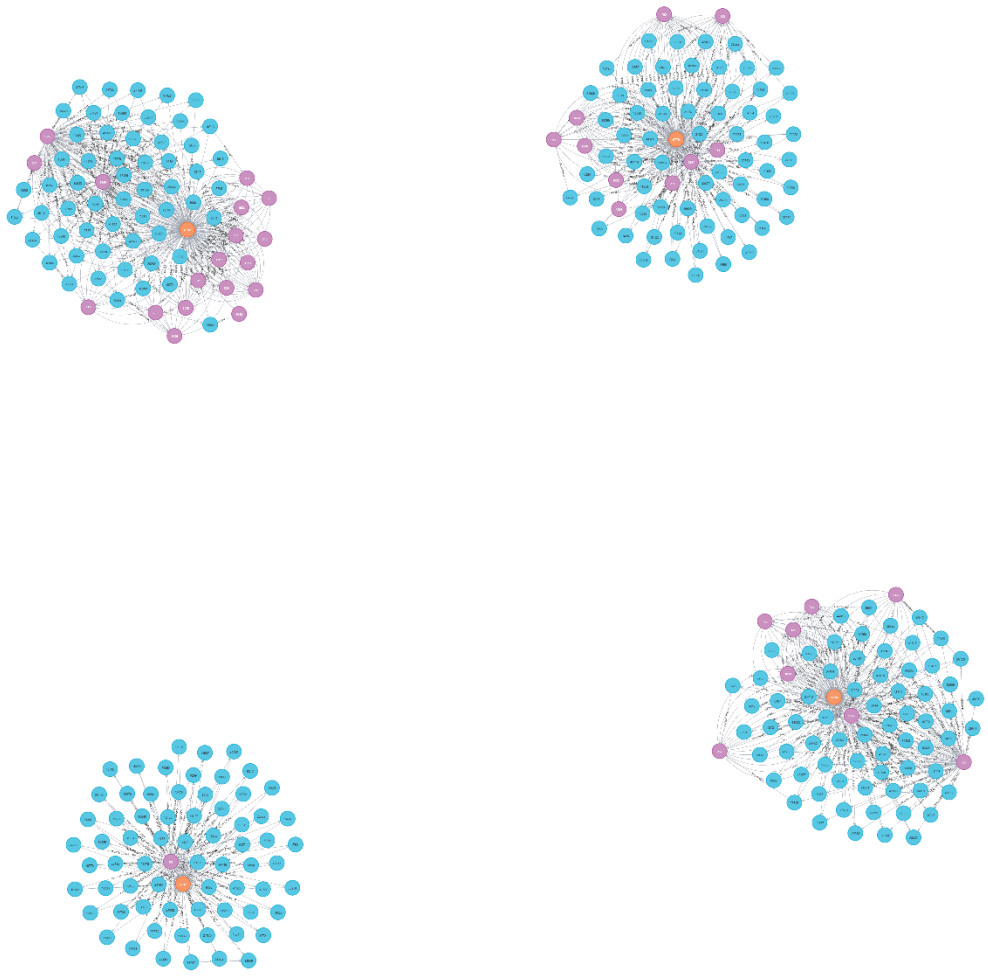
```
// Visualize the lifecycle of chat sessions including creation, participation, and interaction within sessions

MATCH (session:TeamChatSession)

OPTIONAL MATCH (creator:User)-[r1:CREATED]->(session)

OPTIONAL MATCH (user:User)-[r2:JOINS]->(session)

OPTIONAL MATCH (user)-[r3:LEAVES]->(session)
```

```
OPTIONAL MATCH (chatItem:ChatItem)-[r4:PART_OF]->(session)
OPTIONAL MATCH (chatItem)<-[r5:CREATED_CHAT_ITEM]-(user)
OPTIONAL MATCH (chatItem)-[r6:MENTIONED]->(mentioned:User)
OPTIONAL MATCH (response:ChatItem)-[r7:RESPONDS_TO]->(chatItem)
RETURN session, creator, user, chatItem, mentioned, response,
       r1, r2, r3, r4, r5, r6, r7
LIMIT 50;
```

7. Interaction and Participation Analysis between users in a community

Figure 46 shows the different interactions and participation of the users within a network. Multiple interactions such as creation, response, and mentions create this type of comprehensive interaction between users.

Figure 47: Interaction and Participation Analysis



Cypher Query:

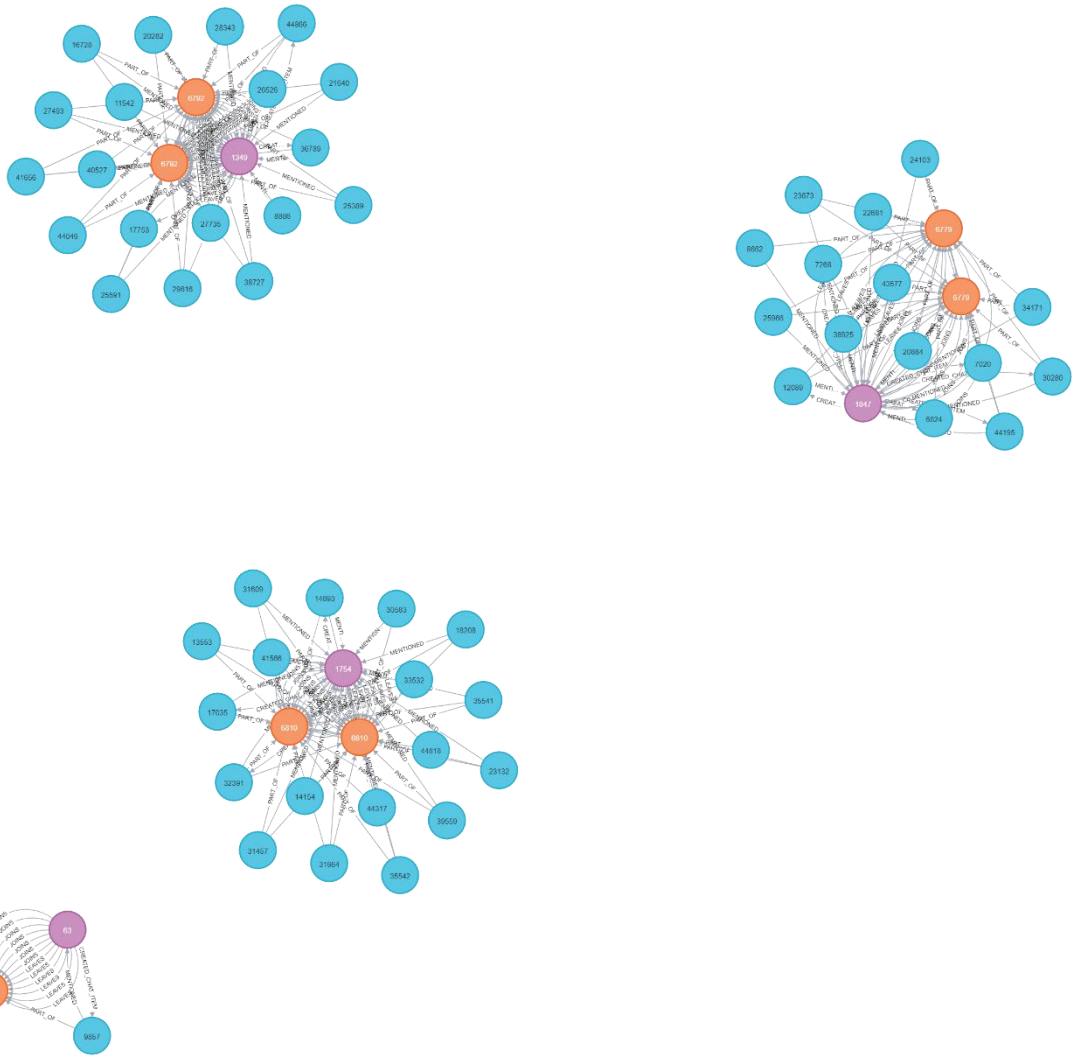
```
// Complex visualization of user interactions across chat sessions including creation, response, and mentions
MATCH (user:User)-[join:JOINS]->(session:TeamChatSession)
OPTIONAL MATCH (user)-[create:CREATED_CHAT_ITEM]->(item:ChatItem)-[:PART_OF]->(session)
OPTIONAL MATCH (item)-[mention:MENTIONED]->(mentionedUser:User)
OPTIONAL MATCH (item)-[response:RESPONDS_TO]->(responseItem:ChatItem),
              (responseUser:User)-[:CREATED_CHAT_ITEM]->(responseItem)
OPTIONAL MATCH (leaver:User)-[leave:LEAVES]->(session)
WITH user, session,
     COLLECT(DISTINCT item) AS createdItems,
     COLLECT(DISTINCT mentionedUser) AS mentionedUsers,
     COLLECT(DISTINCT responseItem) AS responseItems,
     COLLECT(DISTINCT responseUser) AS responseUsers,
     COLLECT(DISTINCT leaver) AS leavers,
     COLLECT(DISTINCT join) AS joins,
     COLLECT(DISTINCT create) AS creates,
     COLLECT(DISTINCT mention) AS mentions,
     COLLECT(DISTINCT response) AS responses,
     COLLECT(DISTINCT leave) AS leaves
RETURN user,
       session,
       createdItems,
       mentionedUsers,
       responseItems,
       responseUsers,
       leavers,
       SIZE(joins) AS joinCount,
       SIZE(creates) AS createCount,
```

```
SIZE(mentions) AS mentionCount,
SIZE(responses) AS responseCount,
SIZE(leaves) AS leaveCount;
```

8. Show the spread of conversation topics through mentions

Figure 48: Spread of Conversation Topics through Mentions

Figure 47 shows the different conversation topics that are spread through chat mentions.



Cypher Query:

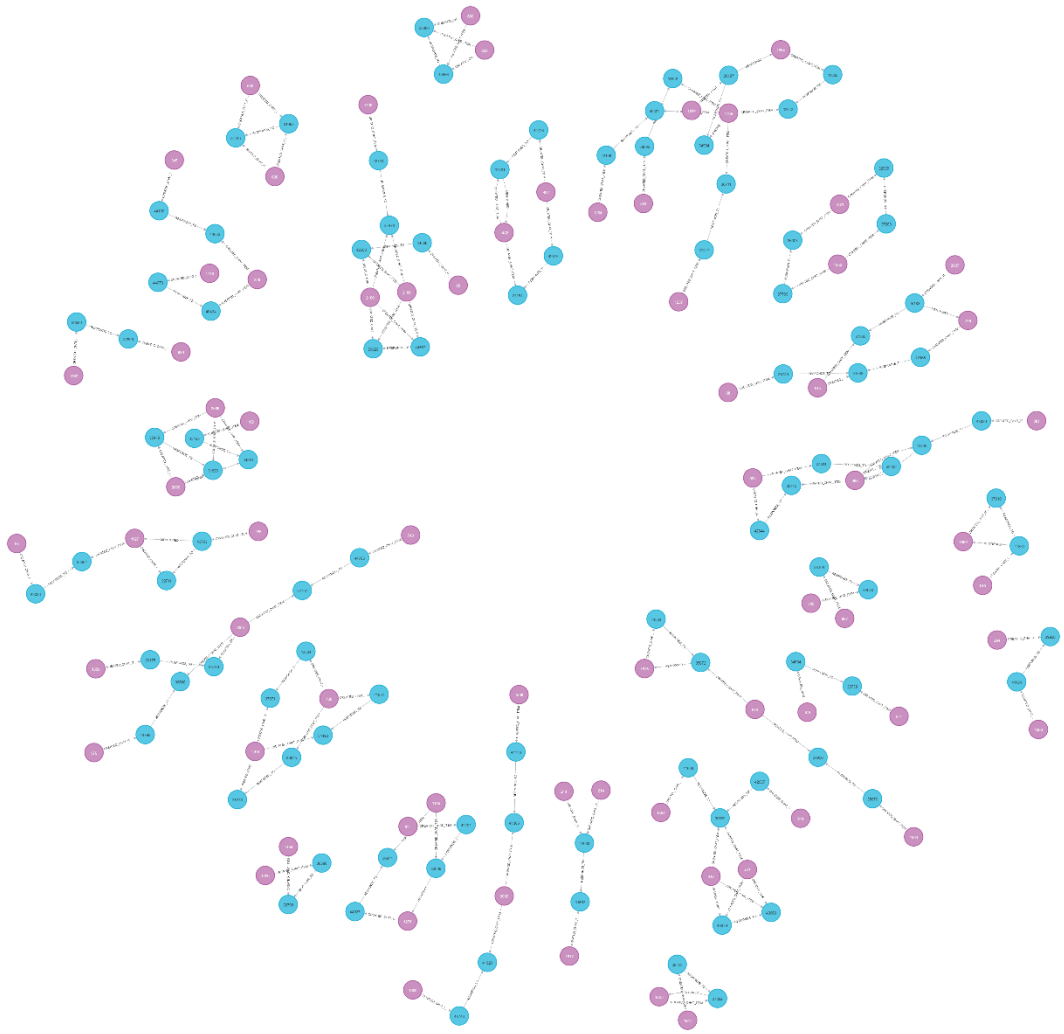
```
MATCH (ci:ChatItem)-[:MENTIONED]->(u:User),
```

```
      (ci)-[:PART_OF]->(t:TeamChatSession)
RETURN t, ci, u
LIMIT 100;
```

9. Visualization of User Connectivity via Mentions and Responses

Figure 48 shows the connections between users via mentions and responses while participating in a chat session.

Figure 49: User Connectivity via Mentions and Responses



Cypher Query:

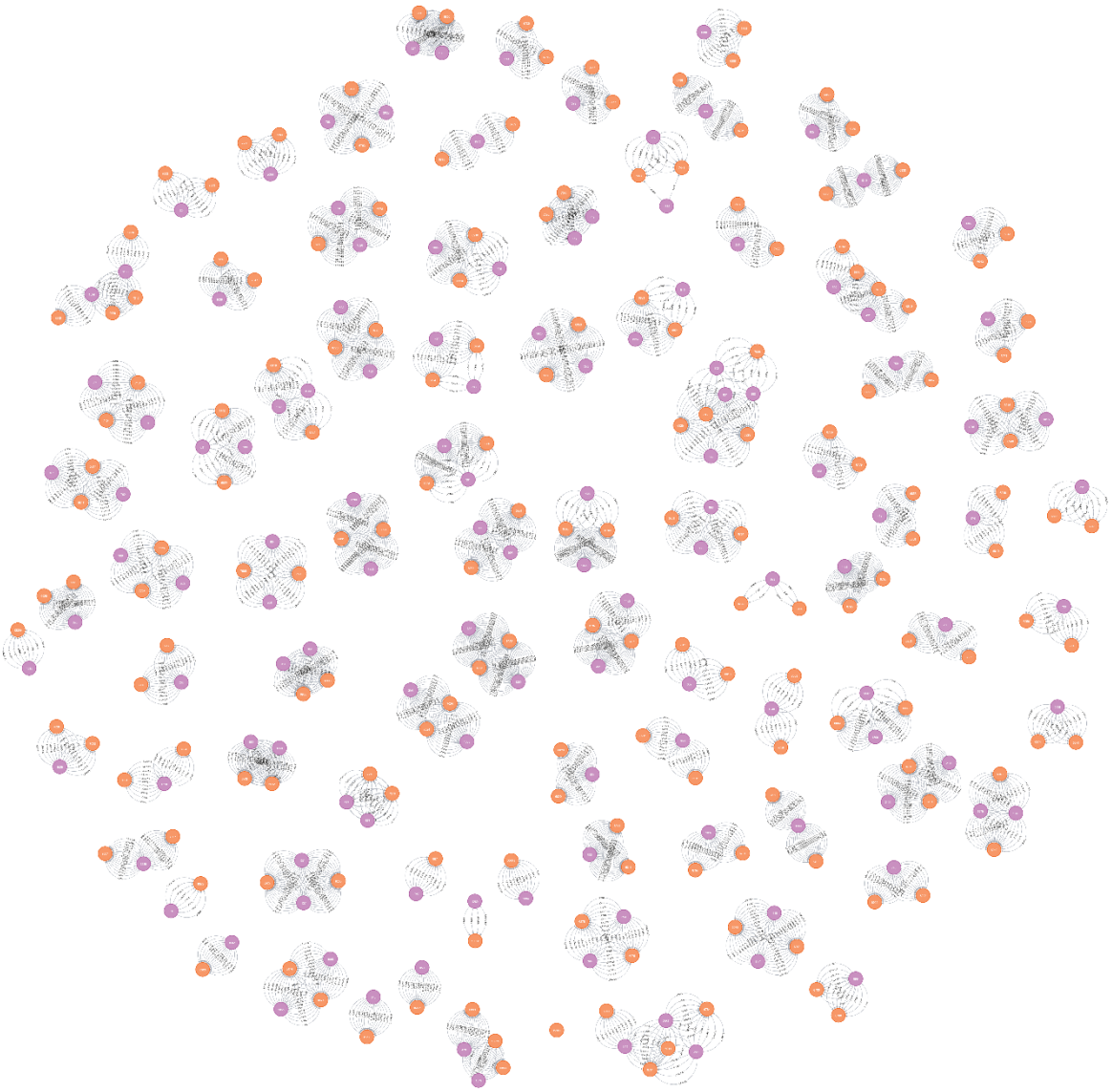
```
MATCH (u:User)-[:CREATED_CHAT_ITEM]->(ci:ChatItem)-[:MENTIONED|RESPONDS_TO]-
>(ci2:ChatItem)<-[:CREATED_CHAT_ITEM]-(u2:User)
```

```
WHERE u <> u2
RETURN u, u2, ci, ci2
LIMIT 50;
```

10. Identify the Central Users in Each Chat Session

Figure 49 identifies the central users in each chat session based on their contributions. The user having top contributions within a graph network is considered a central user.

Figure 50: Identify Central users in Each Chat Session



Cypher Query:

```
MATCH (t:TeamChatSession)<-[:PART_OF]-(c:ChatItem)<-[r:CREATED_CHAT_ITEM]-
(u:User)

WITH t, u, COUNT(r) AS contributions

ORDER BY contributions DESC

WITH t, collect(u)[0] AS topContributor

MATCH (topContributor)-[:JOINS]->(t)

RETURN t AS Session, topContributor AS CentralUser, COUNT(*) AS
InteractionCount;
```

10 Role of Ethics

Data Ethics defines the principle of using data especially personal data fairly and responsibly (Floridi and Taddeo, 2016) as the vast development in the field of Information Technology, Software Systems, Machine Learning, and Internet of Things (IoT) Applications generate data enormously in huge volume, velocity, and variety which is difficult to handle by traditional data processing framework. The issue is very important in medicine and the healthcare field where they collect the records of patient's biomedical information, and personal identity which is very critical in terms of privacy and digital identity. Thus, while storing, managing, and processing those data, if data ethics are not followed properly, it may harm a huge number of individuals and can be the risk of massive privacy and personal data misuse.

So, the main concern of data ethics lies in ensuring those data are maintained ethically including privacy, transparency, and general moral ethics. This is about how data collectors such as data brokers, big companies and governments manage and share data by considering and maintaining privacy and ensuring that the data isn't misused. In addition, as artificial intelligence (AI) and machine learning technologies are developed using big data, the ethics of how this data is used in AI is also a significant topic (Herschel and Miori, 2017).

For example, On September 21, 2012, an incident highlighting the risks of mishandling personal data occurred when a 16-year-old girl in Netherlands accidentally posted her birthday party invitation on Facebook. So, about 3,000 people came to her house for a birthday celebration (Zwitter, 2014). From this example, it is clear that there should be ethical standards and practice for using big data as the discussions on using data ethically have grown socially, politically, and academically (Roche and Jamal, 2021). So, Lawmakers should focus on this topic strictly and introduce the framework and new best practices

In the case of our project "Diabetes Prediction", we used the dataset for modeling different machine learning algorithms and we ensured that there was no personal identification-related data. Thus, making sure the data is used ethically and legally is very important while working on it because it may lead to different privacy-related issues in the people and community which may make people feel less safe in the online environment as their data is not safe.

Therefore, while building data analytics, machine learning, and artificial intelligence projects, we must strictly follow the big data principles including ownership,

transaction transparency, consent, privacy, and openness which is explained in (Malik, 2013). These principles provide a critical framework to guide emerging big data practices. Thus, all companies should focus on implementing the practice of Responsible Data Analytics and Responsible AI. Companies like Microsoft ("Microsoft Responsible AI Standard, v2 GENERAL REQUIREMENTS FOR EXTERNAL RELEASE", 2022), Google ("Responsible Development of AI", n.d.), PWC ("Responsible AI By PWC", n.d.) have already published a framework for AI development and innovation. Organizations have already started adopting the various principles for responsible AI but there is less clarity in the Responsible Big Data use cases.

11 Findings, Limitations & Recommendations

In the project on "diabetes prediction", we successfully executed five different machine learning algorithms including both classification and clustering techniques. Using classification techniques such as logistic regression, we achieved a good accuracy score of 0.82 (Test Score) and 0.83 (Validation Score) as we divided our dataset into three parts: 1. Training (60%), 2. Testing (20%), and 3. Validation (20%) which ensures the reliability and scalability of the model very well. Despite of having a good accuracy score, there is still room for improvement in performing different hyper tuning, data scaling, feature engineering, and testing with other advanced machine learning algorithms which may increase the accuracy of the model. As we are working on the health domain, the accuracy of the model is really important because the model is going to directly impact people's lives.

Similarly, for graph analysis, we loaded the dataset into a neo4j database platform and performed a few data analysis operations. To drive real business value, we have to perform those graph analyses and compare the results with real-world scenarios to drive impact in the business world. As Neo4j graph is widely used by companies and scientists to model different Generative AI and advanced machine learning models, we can perform and apply different machine learning algorithms through the neo4j graph analytics platform to train, model, and solve the actual problem with the help of graph data.

In conclusion, the project implemented big data using Apache Spark and Neo4j to solve real-world business problems such as diabetes prediction and game chat graph analysis respectively which demonstrates how data managing, processing, and analyzing of large volume of data is conducted to gain valuable insights from it.

12 References

Agrawal, D., Das, S. and El Abbadi, A. (2011), "Big data and cloud computing: Current state and future opportunities", *ACM International Conference Proceeding Series*, Association for Computing Machinery, pp. 530–533, doi: 10.1145/1951365.1951432.

Ait Errami, S., Hajji, H., Ait El Kadi, K. and Badir, H. (2023), "Spatial big data architecture: From Data Warehouses and Data Lakes to the LakeHouse", *Journal of Parallel and Distributed Computing*, Academic Press, Vol. 176, pp. 70–79, doi: 10.1016/J.JPDC.2023.02.007.

Al-Sai, Z.A., Abdullah, R. and Husin, M.H. (2019), "Big Data Impacts and Challenges: A Review", *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology, JEEIT 2019 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., pp. 150–155, doi: 10.1109/JEEIT.2019.8717484.

Dos Anjos, J.C.S., Matteussi, K.J., De Souza, P.R.R., Grabher, G.J.A., Borges, G.A., Barbosa, J.L.V., González, G. V., *et al.* (2020), "Data processing model to perform big data analytics in hybrid infrastructures", *IEEE Access*, Institute of Electrical and Electronics Engineers Inc., Vol. 8, pp. 170281–170294, doi: 10.1109/ACCESS.2020.3023344.

Bao, K., Zhang, J., Zhang, Y., Wenjie, W., Feng, F. and He, X. (2023), "Large Language Models for Recommendation: Progresses and Future Directions", *SIGIR-AP 2023 - Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, Association for Computing Machinery, Inc, pp. 306–309, doi: 10.1145/3624918.3629550.

Baştanlar, Y. and Özuysal, M. (2014), "Introduction to Machine Learning", *Methods in Molecular Biology*, Humana Press, Totowa, NJ, Vol. 1107, pp. 105–128, doi: 10.1007/978-1-62703-748-8_7.

Benjelloun, S., Aissi, M.E.M. El, Loukili, Y., Lakhrissi, Y., Ali, S.E. Ben, Chougrad, H. and Boushaki, A. El. (2020), "Big Data Processing: Batch-based processing and stream-based processing", *4th International Conference on Intelligent Computing in Data Sciences, ICDS 2020*, Institute of Electrical and Electronics Engineers Inc., doi: 10.1109/ICDS50568.2020.9268684.

Birhane, A., Kasirzadeh, A., Leslie, D. and Wachter, S. (2023), “Science in the age of large language models”, *Nature Reviews Physics 2023 5:5*, Nature Publishing Group, Vol. 5 No. 5, pp. 277–280, doi: 10.1038/s42254-023-00581-4.

Casado, R. and Younas, M. (2015), “Emerging trends and technologies in big data processing”, *Concurrency and Computation*, article, , Vol. 27 No. 8, pp. 2078–2091, doi: 10.1002/cpe.3398.

“CDC Diabetes Health Indicators - UCI Machine Learning Repository”. (n.d.). , available at: https://www.archive.ics.uci.edu/dataset/891/cdc+diabetes+health+indicators (accessed 13 May 2024).

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., *et al.* (2024), “A Survey on Evaluation of Large Language Models”, *ACM Transactions on Intelligent Systems and Technology*, ACMPUB27New York, NY, doi: 10.1145/3641289.

Chatfield, C. (1986), “Exploratory data analysis”, *European Journal of Operational Research*, North-Holland, Vol. 23 No. 1, pp. 5–13, doi: 10.1016/0377-2217(86)90209-2.

Chen, X.W. and Lin, X. (2014), “Big data deep learning: Challenges and perspectives”, *IEEE Access*, Institute of Electrical and Electronics Engineers Inc., Vol. 2, pp. 514–525, doi: 10.1109/ACCESS.2014.2325029.

Ciu, Z., Damiani, E. and Leida, M. (2007), “Benefits of ontologies in real time data access”, *Proceedings of the 2007 Inaugural IEEE-IES Digital EcoSystems and Technologies Conference, DEST 2007*, pp. 392–397, doi: 10.1109/DEST.2007.372004.

D’Alconzo, A., Drago, I., Morichetta, A., Mellia, M. and Casas, P. (2019), “A survey on big data for network traffic monitoring and analysis”, *IEEE Transactions on Network and Service Management*, Institute of Electrical and Electronics Engineers Inc., Vol. 16 No. 3, pp. 800–813, doi: 10.1109/TNSM.2019.2933358.

“Data growth worldwide 2010-2025 | Statista”. (n.d.). , available at: https://www.statista.com/statistics/871513/worldwide-data-created/ (accessed 15 April 2024).

Ding, X., Chen, L., Emani, M., Liao, C., Lin, P.H., Vanderbruggen, T., Xie, Z., *et al.* (2023), “HPC-GPT: Integrating Large Language Model for High-Performance

Computing", *ACM International Conference Proceeding Series*, Association for Computing Machinery, pp. 951–960, doi: 10.1145/3624062.3624172.

Dittrich, J. and Quiané-Ruiz, J.A. (2012), "Efficient big data processing in Hadoop MapReduce", *Proceedings of the VLDB Endowment*, VLDB EndowmentPUB4722, Vol. 5 No. 12, pp. 2014–2015, doi: 10.14778/2367502.2367562.

Floridi, L. and Taddeo, M. (2016), "What is data ethics?", *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, The Royal Society, Vol. 374 No. 2083, doi: 10.1098/RSTA.2016.0360.

Ghazi, M.R. and Gangodkar, D. (2015), "Hadoop, MapReduce and HDFS: A Developers Perspective", *Procedia Computer Science*, Elsevier, Vol. 48 No. C, pp. 45–50, doi: 10.1016/J.PROCS.2015.04.108.

Goudarzi, M. (2019), "Heterogeneous Architectures for Big Data Batch Processing in MapReduce Paradigm", *IEEE Transactions on Big Data*, Institute of Electrical and Electronics Engineers Inc., Vol. 5 No. 1, pp. 18–33, doi: 10.1109/TBDATA.2017.2736557.

Greeshma, L.. and Pradeepini, G. (2016), "Big Data Analytics with Apache Hadoop MapReduce Framework", *Indian Journal of Science and Technology*, doi: 10.17485/IJST/2016/V9I26/135181.

Guia, J., Soares, V.G. and Bernardino, J. (2017), "Graph Databases: Neo4j Analysis", doi: 10.5220/0006356003510356.

Gurcan, F. and Berigel, M. (2018), "Real-Time Processing of Big Data Streams: Lifecycle, Tools, Tasks, and Challenges", *ISMSIT 2018 - 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies, Proceedings*, Institute of Electrical and Electronics Engineers Inc., doi: 10.1109/ISMSIT.2018.8567061.

"Hadoop framework The Yet Another Resource Negotiator (YARN), which aids... | Download Scientific Diagram". (n.d.). , available at: https://www.researchgate.net/figure/Hadoop-framework-The-Yet-Another-Resource-Negotiator-YARN-which-aids-in-managing_fig6_366018052 (accessed 13 May 2024).

Han, X., Zhang, Z., Ding, N., Gu, Y., Liu, X., Huo, Y., Qiu, J., *et al.* (2021), "Pre-trained models: Past, present and future", *AI Open*, Elsevier, Vol. 2, pp. 225–250, doi: 10.1016/J.AIOPEN.2021.08.002.

Hasani, Z., Kon-Popovska, M. and Velinov, G. (n.d.). "Lambda Architecture for Real Time Big Data Analytic".

Herschel, R. and Miori, V.M. (2017), "Ethics & Big Data", *Technology in Society*, Pergamon, Vol. 49, pp. 31–36, doi: 10.1016/J.TECHSOC.2017.03.003.

Khan, N., Alsaqer, M., Shah, H., Badsha, G., Abbasi, A.A. and Salehian, S. (2018), "The 10 Vs, issues and challenges of big data", *ACM International Conference Proceeding Series*, Association for Computing Machinery, pp. 52–56, doi: 10.1145/3206157.3206166.

Kiran, M., Murphy, P., Monga, I., Dugan, J. and Baveja, S.S. (2015), "Lambda architecture for cost-effective batch and speed big data processing", *Proceedings - 2015 IEEE International Conference on Big Data, IEEE Big Data 2015*, Institute of Electrical and Electronics Engineers Inc., pp. 2785–2792, doi: 10.1109/BIGDATA.2015.7364082.

L'Esteve, R.C. (2023), "Designing a Secure Data Lake", *The Cloud Leader's Handbook*, Apress, Berkeley, CA, pp. 183–201, doi: 10.1007/978-1-4842-9526-7_11.

Liu, X., Lftikhar, N. and Xie, X. (2014), "Survey of real-time processing systems for big data", *ACM International Conference Proceeding Series*, Association for Computing Machinery, pp. 356–361, doi: 10.1145/2628194.2628251.

Londhe, A. and Prasada Rao, P.V.R.D. (2018), "Platforms for big data analytics: Trend towards hybrid era", *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing, ICECDS 2017*, Institute of Electrical and Electronics Engineers Inc., pp. 3235–3238, doi: 10.1109/ICECDS.2017.8390056.

Malik, P. (2013), "Governing Big Data: Principles and practices", *IBM Journal of Research and Development*, IBM, Vol. 57 No. 3/4, pp. 1:1-1:13, doi: 10.1147/JRD.2013.2241359.

Malladi, S., Gao, T., Nichani, E., Damian, A., Lee, J.D., Chen, D. and Arora, S. (2023), "Fine-Tuning Language Models with Just Forward Passes", *Advances in Neural Information Processing Systems*, Vol. 36, pp. 53038–53075.

Meehan, J., Zdonik, S., Tian, S., Tian, Y., Tatbul, N., Dziedzic, A. and Elmore, A. (2016), "Integrating real-time and batch processing in a polystore", *2016 IEEE High Performance Extreme Computing Conference, HPEC 2016*, Institute of Electrical and Electronics Engineers Inc., doi: 10.1109/HPEC.2016.7761585.

"Microsoft Responsible AI Standard, v2 GENERAL REQUIREMENTS FOR EXTERNAL RELEASE". (2022), .

Nandimath, J., Banerjee, E., Patil, A., Kakade, P. and Vaidya, S. (2013), "Big data analysis using Apache Hadoop", *Proceedings of the 2013 IEEE 14th International Conference on Information Reuse and Integration, IEEE IRI 2013*, IEEE Computer Society, pp. 700–703, doi: 10.1109/IRI.2013.6642536.

Nick, T.G. and Campbell, K.M. (2007), "Logistic Regression", *Methods in Molecular Biology (Clifton, N.J.)*, Humana Press, Vol. 404, pp. 273–301, doi: 10.1007/978-1-59745-530-5_14.

Pal, A. and Pal, S.K. (2016), "Pattern recognition: Evolution, mining and big data", *Pattern Recognition and Big Data*, World Scientific Publishing Co. Pte. Ltd., pp. 1–36, doi: 10.1142/9789813144552_0001.

Pathak, A.R., Pandey, M. and Rautaray, S.S. (2020), "Approaches of enhancing interoperations among high performance computing and big data analytics via augmentation", *Cluster Computing*, article, Springer US, New York, Vol. 23 No. 2, pp. 953–988, doi: 10.1007/s10586-019-02960-y.

Pishgoo, B., Akbari Azirani, A. and Raahemi, B. (2021), "A hybrid distributed batch-stream processing approach for anomaly detection", *Information Sciences*, Elsevier, Vol. 543, pp. 309–327, doi: 10.1016/J.INS.2020.07.026.

"Processing paradigms. | Download Scientific Diagram". (n.d.). , available at: https://www.researchgate.net/figure/Processing-paradigms_fig1_266373455 (accessed 13 May 2024).

"Responsible AI By PWC". (n.d.). , available at: https://www.pwc.com/m1/en/services/assurance/risk-assurance/documents/responsible-ai.pdf (accessed 11 May 2024).

"Responsible Development of AI". (n.d.). .

Richards, J.A. (2022), “Supervised Classification Techniques”, *Remote Sensing Digital Image Analysis*, Springer, Cham, pp. 263–367, doi: 10.1007/978-3-030-82327-6_8.

Roche, J. and Jamal, A. (2021), “A Systematic Literature Review of the Role of Ethics in Big Data”, *Advanced Sciences and Technologies for Security Applications*, Springer, Cham, pp. 327–342, doi: 10.1007/978-3-030-68534-8_20.

Saberian, M., Delgado, P. and Raimond, Y. (2019), “Gradient Boosted Decision Tree Neural Network”.

Safaei, A.A. (2017), “Real-time processing of streaming big data”, *Real-Time Systems*, Springer Science and Business Media, LLC, Vol. 53 No. 1, pp. 1–44, doi: 10.1007/S11241-016-9257-0/FIGURES/20.

Sagiroglu, S. and Sinanc, D. (2013), “Big data: A review”, *Proceedings of the 2013 International Conference on Collaboration Technologies and Systems, CTS 2013*, pp. 42–47, doi: 10.1109/CTS.2013.6567202.

Sarker, I.H. (2024), “LLM Potentiality and Awareness: A Position Paper from the Perspective of Trustworthy and Responsible AI Modeling”, *Authorea Preprints*, Authorea, doi: 10.36227/TECHRXIV.170905626.67078570/V1.

Serhani, M.A., El Kassabi, H.T., Taleb, I. and Nujum, A. (2016), “An hybrid approach to quality evaluation across big data value chain”, *Proceedings - 2016 IEEE International Congress on Big Data, BigData Congress 2016*, Institute of Electrical and Electronics Engineers Inc., pp. 418–425, doi: 10.1109/BIGDATACONGRESS.2016.65.

Sinaga, K.P. and Yang, M.S. (2020), “Unsupervised K-means clustering algorithm”, *IEEE Access*, Institute of Electrical and Electronics Engineers Inc., Vol. 8, pp. 80716–80727, doi: 10.1109/ACCESS.2020.2988796.

Sinanc Terzi, D., Demirezen, U. and Sagiroglu, S. (2016), “EVALUATIONS OF BIG DATA PROCESSING”, *Services Transactions on Big Data*, Vol. 3 No. 1, p. 44.

Singh, K.N., Behera, R.K. and Mantri, J.K. (2019), “Big data ecosystem: Review on architectural evolution”, *Advances in Intelligent Systems and Computing*, Springer Verlag, Vol. 813, pp. 335–345, doi: 10.1007/978-981-13-1498-8_30/COVER.

singh, N. and Gaur, B. (2019), "Data Preprocessing: A Step-by-Step Guide for Clean and Usable Data", *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, Ninety Nine Publication, Vol. 10 No. 2, pp. 1148–1153, doi: 10.61841/TURCOMAT.V10I2.14384.

Sisodia, D. and Singh, L. (n.d.). "Clustering Techniques: A Brief Survey of Different Clustering Algorithms".

Speiser, J.L., Miller, M.E., Tooze, J. and Ip, E. (2019), "A comparison of random forest variable selection methods for classification prediction modeling", *Expert Systems with Applications*, Pergamon, Vol. 134, pp. 93–101, doi: 10.1016/J.ESWA.2019.05.028.

Syed, A.R., Gillela, K. and Venugopal, C. (2013), "The Future Revolution on Big Data", *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 2.

Vassakis, K., Petrakis, E. and Kopanakis, I. (2018), "Big data analytics: Applications, prospects and challenges", *Lecture Notes on Data Engineering and Communications Technologies*, Springer Science and Business Media Deutschland GmbH, Vol. 10, pp. 3–20, doi: 10.1007/978-3-319-67925-9_1/COVER.

Velleman, P.F. and Hoaglin, D.C. (2012), "Exploratory data analysis.", *APA Handbook of Research Methods in Psychology, Vol 3: Data Analysis and Research Publication.*, American Psychological Association, Washington, pp. 51–70, doi: 10.1037/13621-003.

Volk, M., Hart, S., Bosse, S. and Turowski, K. (n.d.). "PREPRINT A Big Data Classification Framework for IT Projects How much is Big Data? A Classification Framework for IT Projects and Technologies Indicate Submission Type: Full Papers".

Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., *et al.* (2023), "A Survey on Large Language Model based Autonomous Agents", *Frontiers of Computer Science 2024 18:6*, Springer, Vol. 18 No. 6, pp. 1–26, doi: 10.1007/S11704-024-40231-1/METRICS.

Yaqoob, I., Hashem, I.A.T., Gani, A., Mokhtar, S., Ahmed, E., Anuar, N.B. and Vasilakos, A. V. (2016), "Big data: From beginning to future", *International*

Journal of Information Management, Pergamon, Vol. 36 No. 6, pp. 1231–1247, doi: 10.1016/J.IJINFOMGT.2016.07.009.

Yu, D. and Deng, L. (2015), "Gaussian Mixture Models", Springer, London, pp. 13–21, doi: 10.1007/978-1-4471-5779-3_2.

Zhang, M., He, J., Lei, S., Yue, M., Wang, L. and Lu, C.-T. (2024), "Can LLM Find the Green Circle? Investigation and Human-Guided Tool Manipulation for Compositional Generalization", *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 11996–12000, doi: 10.1109/ICASSP48485.2024.10446355.

Zhang, Y. (2010), "Types of Machine Learning Algorithms", *New Advances in Machine Learning*, IntechOpen, pp. 19–25.

ZhaoHaiyan, ChenHanjie, YangFan, LiuNinghao, DengHuiqi, CaiHengyi, WangShuaiqiang, *et al.* (2024), "Explainability for Large Language Models: A Survey", *ACM Transactions on Intelligent Systems and Technology*, ACMPUB27New York, NY, Vol. 15 No. 2, pp. 1–38, doi: 10.1145/3639372.

Zheng, T., Chen, G., Wang, X., Chen, C., Wang, X. and Luo, S. (2019), "Real-time intelligent big data processing: technology, platform, and applications", *Science China Information Sciences*, Science in China Press, Vol. 62 No. 8, pp. 1–12, doi: 10.1007/S11432-018-9834-8/METRICS.

Zwitter, A. (2014), "Big Data ethics", *Http://Dx.Doi.Org/10.1177/2053951714559253*, SAGE PublicationsSage UK: London, England, Vol. 1 No. 2, doi: 10.1177/2053951714559253.

13 Appendices

13.1 A0

The “Catch the Pink Flamingo” dataset is hosted in my GitHub repository as follows:

https://github.com/DineshThapaX/neo4j_sample_project

This repository consists of 6 different files of chat data from Pink Flamingo dataset which is used for the graph analysis using Neo4j in this project.

13.2 A1

The source code of the exploratory data analysis, machine learning techniques implementation are included in the dinesh_source_code.zip file. It includes: -

1. EDA_DineshThapa.ipynb: For Exploratory Data Analysis
2. Classification_DineshThapa.ipynb: For Classification Techniques (Logistic Regression, Random Forest and Gradient Boosted Tree Classifier)
3. KMeans_DineshThapa.ipynb: For K-means clustering
4. Gaussian_GMM_DineshThapa.ipynb: For Gaussian Mixture Model (GMM)