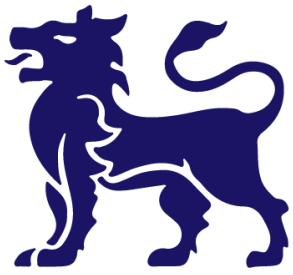
CMP 7203: Big Data Management

Final Assessment

Big Data Ecosystem

Sachin

Student Id: 23235298



BIRMINGHAM CITY University

Faculty of Computing, Engineering, and The Built Environment

School of Computing and Digital Technology

Table of Contents

1. Introduction to Big Data

In the early 2000s, Big Data emerged, emphasizing Volume, Velocity, and Variety. Over time, Veracity and Value were added, highlighting data credibility and actionable insights. While discussions include more dimensions like 8Vs or 10Vs, the focus remains on the core five: Volume, Velocity, Variety, Veracity, and Value. These aspects are vital for comprehending and utilizing Big Data effectively.

1.1 Vs in Big Data

1. **Volume**: Volume refers to the vast quantity of data, continually expanding across every sector. This sheer magnitude of data offers significant potential for improved predictive capabilities in the future.
2. **Velocity**: Velocity refers to the speed at which data can be processed for decision-making purposes. With data arriving in large volumes at an ever-increasing rate, it's crucial to extract valuable insights in real time for effective analysis.
3. **Variety**: Variety spans a wide spectrum of data origins, spanning structured, unstructured, and semi-structured formats. These origins encompass textual content, sensor-generated data, audio and visual recordings, as well as graphical representations. This broad range of data sources facilitates a thorough exploration of big data, harnessing the depth of insights derived from various origins.
4. **Veracity**: Veracity refers to the reliability and consistency of data, which can vary significantly in terms of coverage, accuracy, and timeliness.
5. **Value**: Value refers to the ability of individuals or organizations to translate large datasets into practical advantages. This process entails more than just gathering data; it also involves utilizing it efficiently to achieve particular goals (Saeed, 2021)

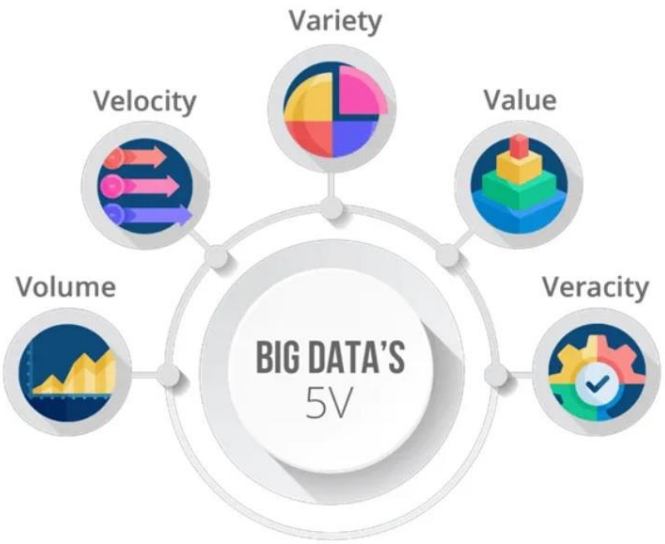


Figure 1: Five Vs of Big Data

1.2 Big Data Processing Paradigm

Different processing methods have evolved: Batch, Real-time, and the recent Hybrid approach. Batch handles data in groups, Real-time processes data immediately, and Hybrid combines both for efficiency. Each caters to distinct needs and technological advancements.

1.2.1 Batch Processing

Batch processing efficiently handles large volumes of static data without considering new data during processing. It scales via parallel distributed processing frameworks like MapReduce, offering simplicity and scalability but struggles with iterative tasks and real-time data. Recent implementations aim to overcome these limitations. Despite reliability, it may not suit low-latency applications or accommodate new data flexibly. Examples include analyzing website logs for customer behavior patterns. Batch Big Data analytics find applications in social networks, graph mining, scientific research, and more as shown in Figure 2.

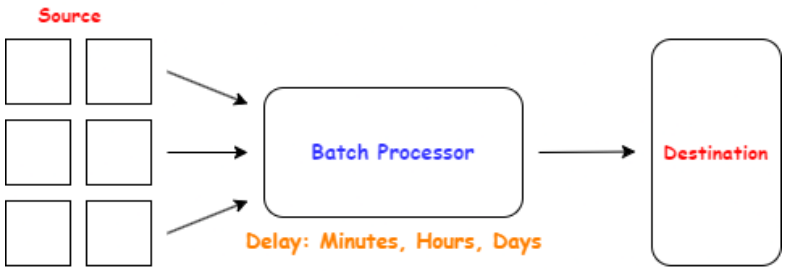


Figure 2: Batch Processing

1.2.2 Real-time Processing

Real-time processing analyses streaming data with minimal latency, akin to continuous small batch processing, storing data in memory for swift analysis. It shares principles with batch processing, focusing on distribution and parallelism. This approach, exemplified by identifying trending topics on Twitter, find applications in managing transportation, energy, waste in smart cities, disaster management, production optimization with sensor data, and entertainment platform analytics. Real-time processing addresses the velocity of Big Data, crucial for diverse domains where timely insights from heterogeneous data streams are imperative as shown in Figure 3.

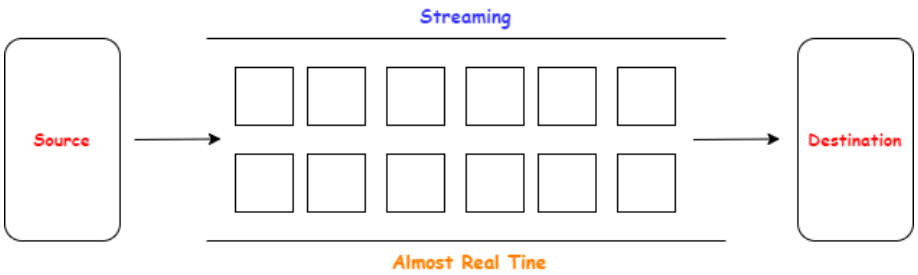


Figure 3: Real-Time Processing

1.2.3 Hybrid Model

Numerous fields of application require combining batch and real-time processing approaches, which is accomplished through a hybrid framework referred to as the Lambda Architecture. This architecture comprises three layers as shown in Figure 4.

- **Batch layer**, also known as batch processing, oversees the primary dataset, which remains unchanged and is stored within a distributed file system
- **Serving layer**, responsible for batch results, loads and presents batch views in a datastore, enabling seamless querying
- **Speed layer**, dedicated to real-time processing, concentrates on managing incoming data with stringent low-latency demands.

In order to obtain thorough outcomes, it is necessary to query both batch and real-time views and merge their results. This amalgamation process, which involves synchronization and addresses various complex issues, is handled by the Combination layer (Casado, 2015).

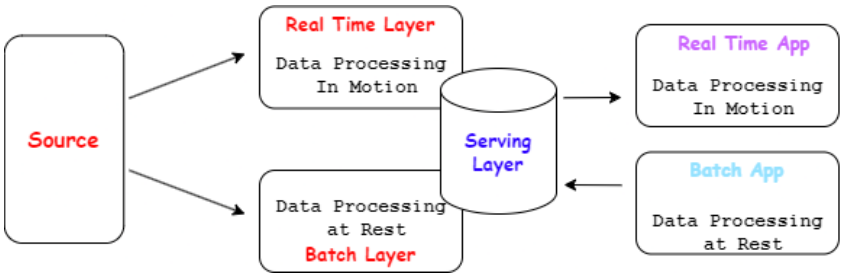


Figure 4: Hybrid Processing

2. Different organizations and use cases using big data paradigms.

Companies Names	Batch Processing	Real-Time Processing	Hybrid Processing
Amazon	Inventory management, order fulfillment, and financial reporting.	Real-time recommendation engine for personalized product suggestions.	Fraud detection and real-time inventory updates.
Google	Search indexing, and data analytics.	YouTube processes live video streams.	AdWords platform for ad campaign optimization.
Facebook	Data warehousing, analytics.	Real-time notifications and news feed.	User engagement analytics.
Netflix	Content recommendation algorithms.	Real-time video streaming service.	Personalized content delivery.
Uber	Financial reporting, driver payouts.	Real-time ride tracking and surge pricing.	Route optimization.
Twitter	Trend analysis, user activity metrics.	Real-time tweet delivery and notifications.	Sentiment analysis.
Airbnb	Pricing optimization	Real-time booking system and notifications.	Dynamic pricing
Salesforce	Data synchronization, reporting	Real-time sales alerts and lead scoring	Customer insights
Spotify	Music recommendation algorithms	Real-time music streaming service	Personalized playlists
Microsoft	Data warehousing, large-scale data transformations	Real-time data ingestion and processing (Azure Stream Analytics)	Event-driven applications and analytics.
IBM	Mainframe systems (e.g., payroll, billing)	Real-time data analytics (IBM Streams)	Data integration and analytics solutions
Oracle	Data loading, ETL, reporting (Oracle Database)	Real-time data processing (Oracle Stream Analytics)	Complex event processing and real-time analytics
Apple	Batch processing for app store analytics	Real-time notifications and updates for app users	Hybrid approach for personalized recommendations
Tesla	Batch processing for vehicle diagnostics and software updates.	Real-time monitoring of vehicle performance and safety.	Hybrid approach for autonomous driving algorithms.

Table 1: Organizations using big data paradigms

3. Future role of big data in the era of LLMs

In the rapidly evolving landscape of artificial intelligence, the synergy between big data and Large Language Models (LLMs) is poised to redefine the future of AI-driven applications. As LLMs continue to advance, fueled by extensive training datasets and innovative methodologies, the role of big data becomes increasingly pivotal. This convergence holds the promise of unlocking new frontiers in natural language processing, code comprehension, and beyond. In this discussion, we explore the transformative potential and emerging roles of big data in shaping the era of LLMs.

3.1 BitNet b1.58 and the Dawn of Efficient 1-Bit Large Language Models

BitNet b1.58, a 1-bit LLM, offers performance akin to full-precision models but with enhanced efficiency in latency, memory, and energy usage. It signals a shift towards cost-effective, scalable LLMs capable of handling extensive data with less computational demand. Organizations can boost NLP tasks, streamline model deployment, and unlock data-driven insights with 1-bit LLMs, advancing AI integration with big data analytics for innovative applications (Ma, 2024)

3.2 The Influence of Big Data on Large Language Models (LLMs)

Big data significantly shapes Large Language Models (LLMs) in AI, enabling adaptability through extensive training datasets. Despite challenges like bias mitigation and scalability, big data fosters emerging applications such as multimodal understanding and performance enhancement in low-resource languages. It fundamentally influences LLMs' trajectory and potential impact, offering avenues for innovation and progress in AI-driven language understanding and generation across diverse domains (Zhou, 2024).

3.3 Transforming AI Mathematics: The Fusion of Big Data and LLM

The integration of Big Data with Large Language Models (LLMs) to enhance Human-Computer Mutual Assistance (HCMA) involves combining Mathematical Sequencing and Positioning Systems (MSPS) with LLMs. This process aims to convert the opaque nature of machine learning into a transparent one for the advancement of AI mathematics. Utilizing Big Data, LLMs are trained on extensive text datasets, empowering them to produce new text closely resembling the training corpus. This fusion elevates the capabilities of LLMs in natural language processing tasks, fostering progress in AI technology (Zou, 2023).

3.4 Assessing Large Language Models' Code Reading Abilities in Big Data

Assessing Large Language Models' (LLMs) code comprehension in Big Data via metamorphic testing reveals strengths and limitations. By subjecting LLMs to diverse inputs, including modified code snippets, weaknesses are exposed, enhancing their utility in pivotal coding scenarios. Utilizing metamorphic relations, varied test cases challenge LLMs' understanding and generation of code, ensuring robustness in complex Big Data environments. This research illuminates LLMs' role in navigating Big Data intricacies, bolstering their efficacy in generating and comprehending code within dynamic data landscapes, vital for real-world applications (Li, 2023).

4. Introduction to the Big Data Ecosystem

As shown in Figure 5 Big Data ecosystem was established using the Catch the Pink Flamingo dataset. The initiative commenced with an exploratory data analysis (EDA) to understand dataset characteristics. Subsequently, classification techniques such as Decision Trees and Support Vector Machines were applied for effective data grouping. Post-classification, clustering methodologies, including K-Means and Gaussian Mixture Models were utilized to reveal underlying patterns within the dataset. Lastly, Neo4j was leveraged for graph analytics, enhancing comprehension of data interconnections and relationships. This comprehensive strategy aims to extract valuable insights and facilitate informed decision-making processes from the Catch the Pink Flamingo dataset.

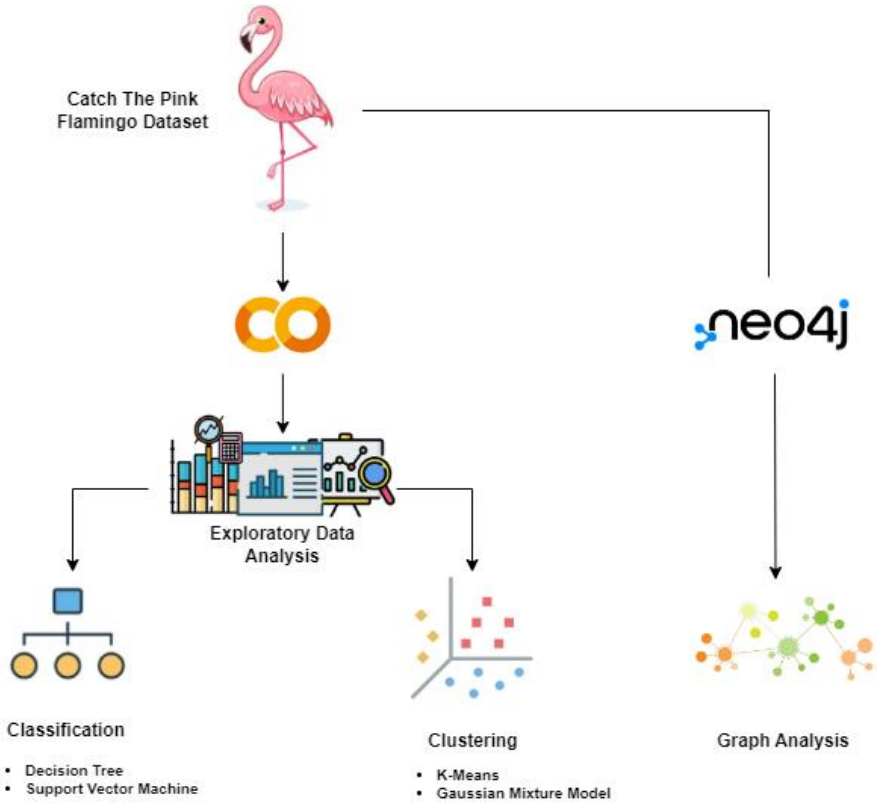


Figure 5: Complete Flow Chart of Big Data Ecosystem

5. Catch The Pink Flamingo

Catch the Pink Flamingo, developed by Eglence Inc., is a multiplayer game where players aim to capture Pink Flamingos across varying maps. Players progress through levels, facing increasingly challenging missions and larger maps.

The game encourages teamwork, allowing players to join or create teams. Each level introduces new missions and larger maps, requiring players to adapt and strategize. Players communicate through team chat boards and social media platforms like Twitter.

Level 1 serves as a tutorial, providing a foundation for gameplay. Successful completion of missions earns points while capturing the wrong flamingo results in point deductions. The dynamic gameplay and interactive features make Catch the Pink Flamingo an engaging multiplayer experience.

6. Data Description

Out of the 15 dataset files, 9 will be used for exploratory data analysis (EDA), labelled *ad-clicks.csv, buy-clicks.csv, users.csv, team.csv, team- assignments.csv, level-events.csv, user-session.csv, game-clicks.csv, and combined_ data.csv, discussed in Table 2.*

The remaining 6 files, labelled *chat_create_team_chat.csv, chat_item_team_chat.csv, chat_join_team_chat.csv, chat_leave_team_chat.csv, chat_mention_team_chat.csv, and chat_respond_team_chat.csv* will be used for graph analysis, discussed in Table 3.

File Name	Description	Fields
ad-clicks.csv	A line is added to this file when a player clicks on an advertisement in the Flamingo app.	**timestamp:** When the click occurred. **txId:** A unique id (within ad-clicks.log) for the click. **userSessionid:** The id of the user session for the user who made the click. **teamid:** The current team id of the user who made the click. **userid:** The user id of the user who made the click. **adId:** The id of the ad clicked on. **adCategory:** The category/type of ad clicked on.
buy-clicks.csv	A line is added to this file when a player makes an in-app purchase in the Flamingo app.	**timestamp:** When the purchase was made. **txId:** A unique id (within buy-clicks.log) for the purchase. **userSessionId:** The id of the user session for the user who made the purchase. **team:** The current team id of the user who made the purchase. **userId:** The user id of the user who made the purchase. **buyId:** The id of the item purchased. **price:** The price of the item purchased.
users.csv	This file contains a line for each user playing the game.	**timestamp:** When user first played the game. **userId:** The user id assigned to the user.

		nick: The nickname chosen by the user. **twitter:** The twitter handle of the user. **dob:** The date of birth of the user. **country:** The two-letter country code where the user lives.
team.csv	This file contains a line for each team terminated in the game.	**teamId:** The id of the team **name:** The name of the team. **teamCreationTime:** The timestamp when the team was created. **teamEndTime:** The timestamp when the last member left the team. **strength:** A measure of team strength, roughly corresponding to the success of a team. **currentLevel:** The current level of the team.
team- assignments.csv	A line is added to this file each time a user joins a team. A user can be in at most a single team at a time.	**timestamp:** When the user joined the team. **team:** The id of the team. **userId:** The id of the user. **assignmentId:** A unique id for this assignment.
level-events.csv	A line is added to this file each time a team starts or finishes a level in the game	**timestamp:** When the event occurred. **eventId:** a unique id for the event. **teamId:** the id of the team. **teamLevel:** the level started or completed. **eventType:** the type of event, either start or end.
user- session.csv	Each line in this file describes a user session, which denotes when a user starts and stops playing the game. Additionally, when a team goes to the next level in the game, the session is ended for each user in the team and a new one started.	**timestamp:** a timestamp denoting when the event occurred. **userSessionId:** a unique id for the session. **userId:** the current user's ID. **teamId:** the current user's team. **assignmentId:** the team assignment id for the user to the team.

		sessionType: whether the event is the start or end of a session. **teamLevel:** the level of the team during this session. **platformType:** the type of platform of the user during this session.
game-clicks.csv	A line is added to this file each time a user performs a click in the game.	**timestamp:** when the click occurred. **clickId:** a unique id for the click. **userId:** the id of the user performing the click. **userSessionId:** the id of the session of the user when the click is performed. **isHit:** denotes if the click was on a flamingo (value is 1) or missed the flamingo (value is 0). **teamId:** the id of the team of the user. **teamLevel:** the current level of the team of the user.
combined_ data.csv	Combines data from 3 of the log files: user-session.csv, buy-clicks.csv, and game-clicks.csv.	**userid**: User ID **userSessionid:** User session ID **team_level:** User's team level **platformType:** Platform used by user **count_gameclicks:** Total number of game clicks for user session **count_hits:** Total number of game hits for user session **count_buyid:** Total number of purchases for user session **avg_price:** Average purchase price for user session

Table 2: Game Dataset Description

File Name	Description	Fields
chat_create_team_chat.csv	A line is added to this file when a player creates a new chat with their team.	userid, teamid, TeamChatSessionID, timestamp
chat_item_team_chat.csv	Creates nodes labeled ChatItems.	userid, teamchatsessionid, chatitemid, timestamp
chat_join_team_chat.csv	Creates an edge labeled “Joins” from User to TeamChatSession.	userid, TeamChatSessionID, timestamp
chat_leave_team_chat.csv	Creates an edge labeled “Leaves” from User to TeamChatSession.	userid, teamchatsessionid, timestamp
chat_mention_team_chat.csv	Creates an edge labeled “Mentioned”.	ChatItem, userid, timestamp
chat_respond_team_chat.csv	A line is added to this file when player with chatid2 responds to a chat post by another player with chatid1.	chatid1, chatid2, timestamp

Table 3: Chat Dataset Description

7. Exploratory Data Analysis

Exploratory Data Analysis means looking closely at data in the beginning to find patterns, and unusual things, test ideas, and make sure our guesses are right. We do this by using numbers and graphs to summarize the information (Good, 1983).

7.1 Ad-Clicks

7.1.1 Are there any missing values?

There are no missing values present in the dataset as shown in Figure 6.

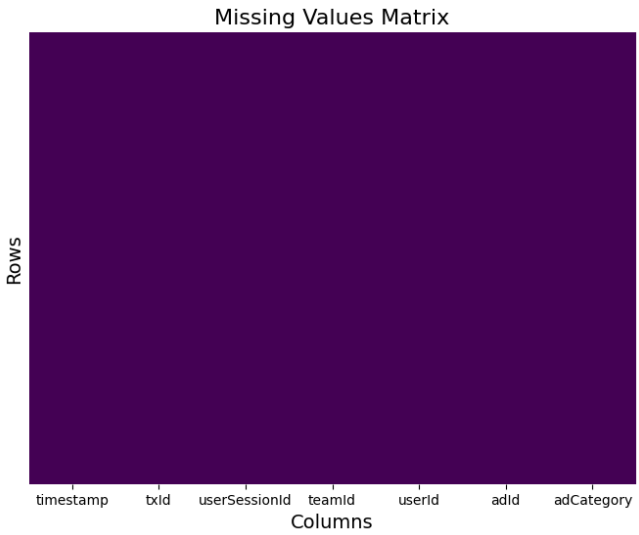


Figure 6: Missing Values In ad-clicks Dataset

7.1.2 Ad Click Trends Over Time?

The time series plot displays daily counts of ad clicks over time. It illustrates fluctuations in user engagement with advertisements, aiding in understanding trends and informing strategies for optimizing ad placement and enhancing user interaction within the app as shown in Figure 7.

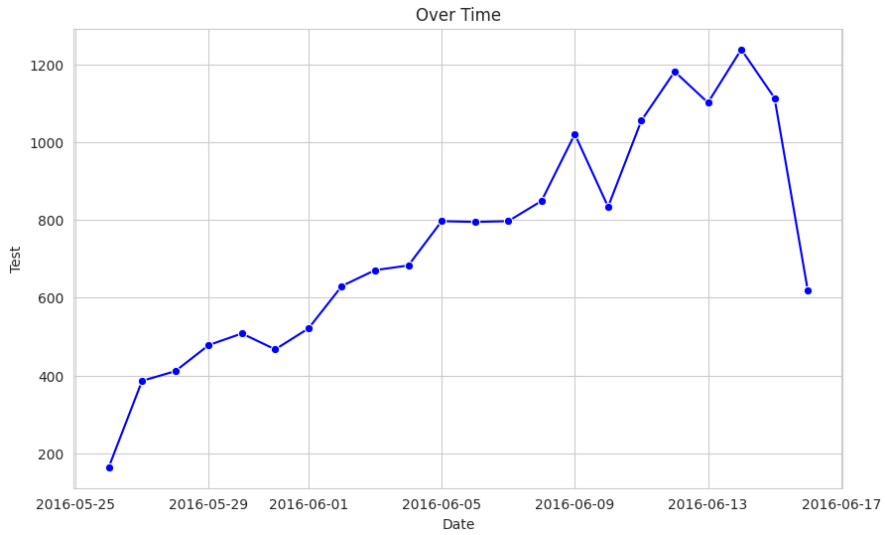


Figure 7: Depicting the daily counts of ad clicks over time.

7.1.3 Which Teams Dominate Ad Clicks?

The largest block in the tree map represents Team 64, boasting a substantial count of 684 ad clicks. This visualization highlights the distribution of ad click counts across the top 10 teams, with Team 64 emerging as the leader in user engagement with advertisements as shown in Figure 8.

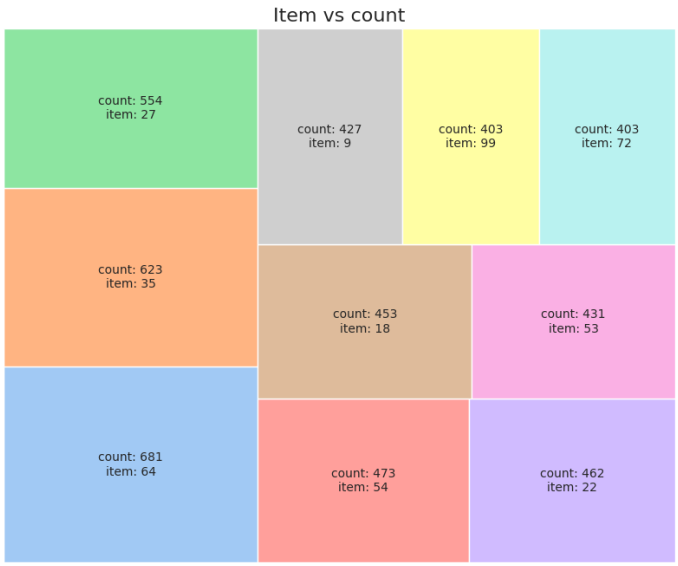


Figure 8: Ad click distribution among top teams

7.1.4 Distribution of Ad Categories Across ad-Ids

The scatter plot illustrates a notable concentration of ad-Ids associated with the 'Computers' and 'Games' categories, indicating a higher frequency of ads belonging to these categories. This observation suggests potential user interest or targeted advertising strategies within these popular categories as shown in Figure 9.

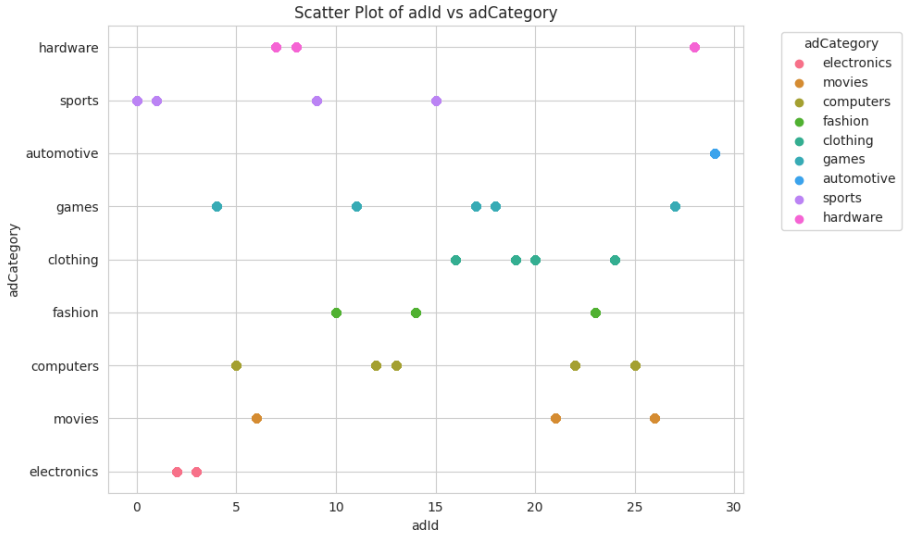


Figure 9: Distribution of ad Categories

7.1.5 Distribution of Session Durations

Histogram depicting the distribution of session durations in minutes for users. Each bar represents a range of session durations, with colors indicating varying frequencies of sessions falling within each range as shown in Figure 10.

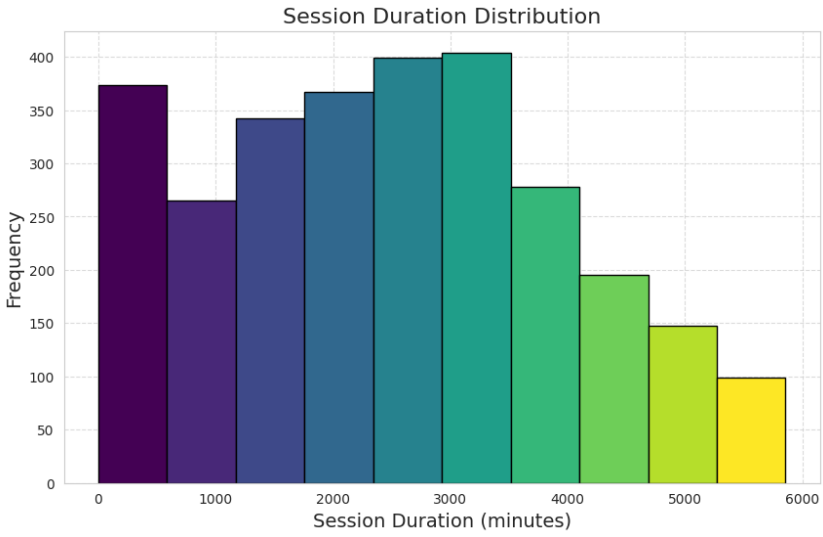


Figure 10: Session Duration Distribution

7.1.6 Distribution of Ad Activity

This bar plot unveils that ad-id 20 garners the highest activity, surpassing 600 clicks. This insight sheds light on the most engaging advertisement, suggesting its effectiveness or popularity among users as shown in Figure 11.

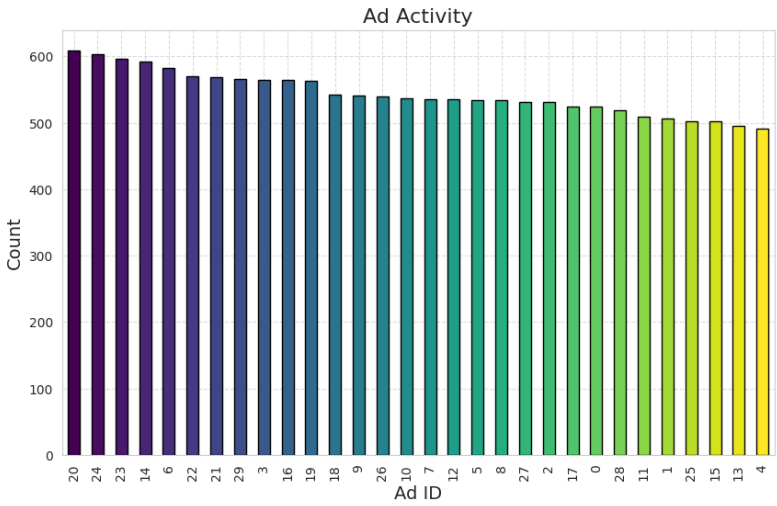


Figure 11: Ad Activity Distribution

7.1.7 Distribution of Ad Categories

Bar plot showcasing the distribution of ad categories. Each bar represents a unique ad category, with colors denoting different categories. The height of each bar corresponds to the count of ads within the respective category, providing insights into the diversity and prevalence of ad types. Observing the bar plot, it's evident that the 'Computers' and 'Games' categories stand out, each recording over 2500 counts as shown in Figure 12.

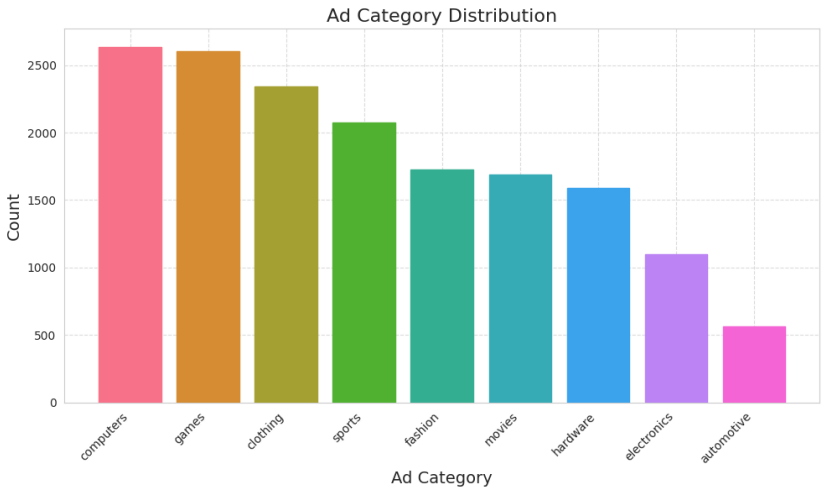


Figure 12: Ad Category Distribution

7.2 Buy-Clicks

7.2.1 Are there any missing values?

There are no missing values present in the dataset as shown in Figure 13.

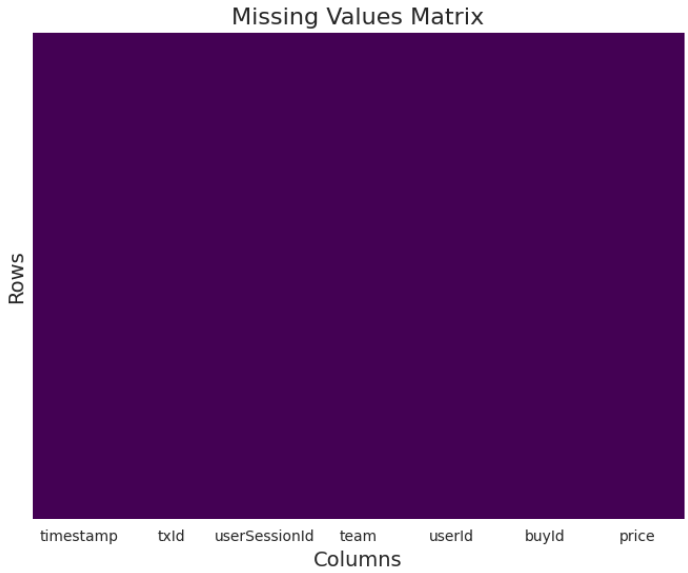


Figure 13: Missing Values in Buy-Clicks Dataset

7.2.2 Daily Counts of In-App Purchases Over Time

The time series plot displays daily in-app purchase counts, revealing fluctuations over time. It aids in assessing user engagement levels and identifying periods of increased or decreased purchase activity, prompting further investigation into influencing factors as shown in Figure 14.

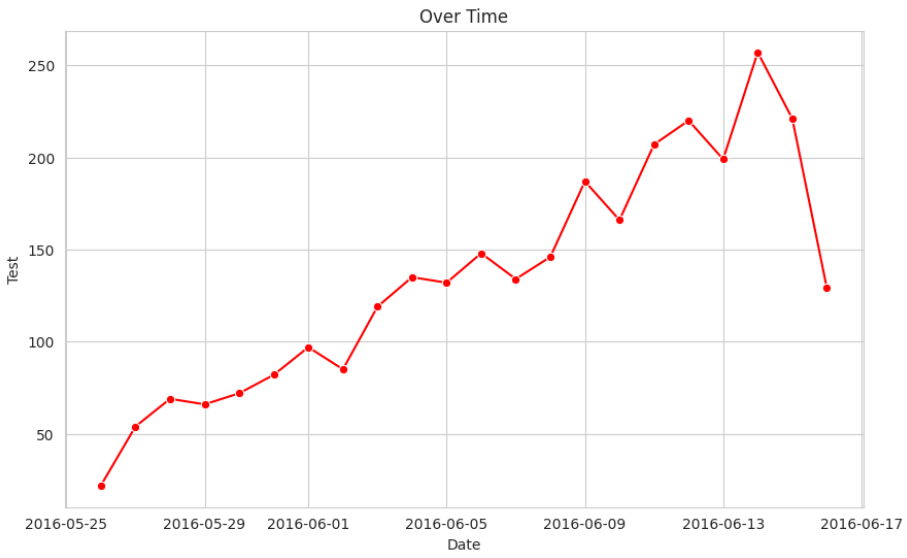


Figure 14: Depicting the daily counts of buy-clicks over time.

7.2.3 Top 20 Teams with the Highest Purchase Counts

By examining the histogram, we can identify the top 20 teams with the highest purchase counts. This analysis provides insights into the purchasing behavior of different teams within the app, highlighting potential areas for targeted marketing or incentives to encourage further purchases. Revealing Team 27 with over 100 purchases and Team 64 with approximately 100 purchases as shown in Figure 15.

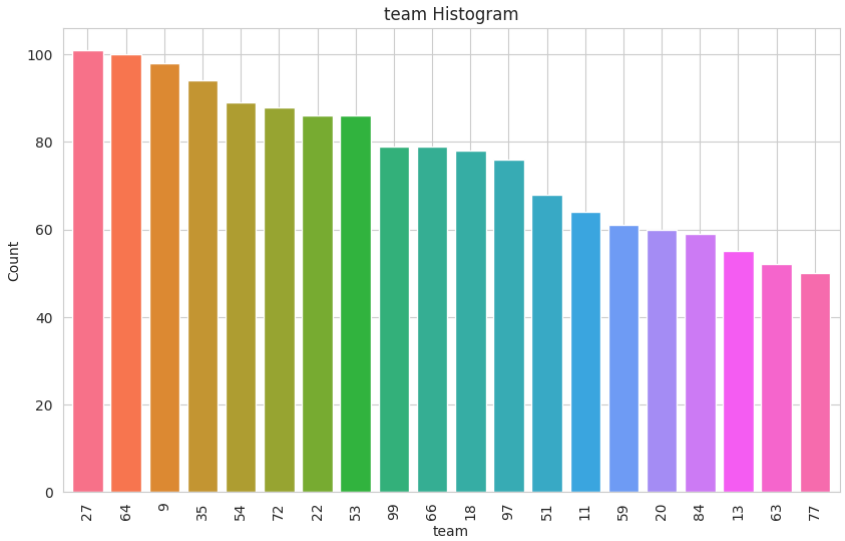


Figure 15: Top team's purchase counts distribution

7.2.4 Correlation Analysis

The correlation plot illustrates the relationship between the team, buy ID, and price. Notably, the team shows no significant correlation with other variables. However, buy ID and price exhibit a strong positive correlation, corroborated by findings in the Combined Data section as shown in Figure 16.

Figure 16: Correlation Analysis: Team, Buy ID, and Price

7.3 Game-Clicks

7.3.1 Are there any missing values?

There are no missing values present in the dataset as shown in Figure 17.

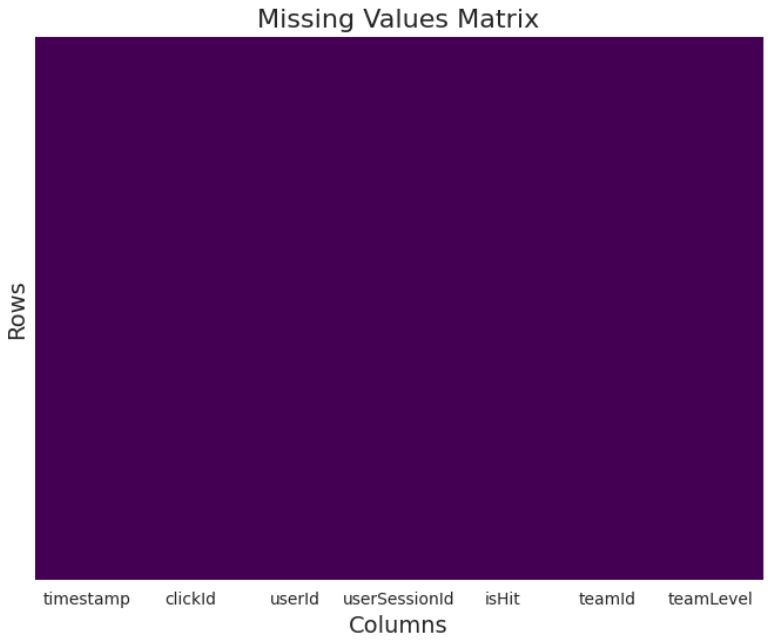


Figure 17: Missing Values in Game-Clicks Dataset

7.3.2 Daily Click Counts in the Game Over Time

The time series plot illustrates daily click counts in the game, revealing fluctuations over time. It offers insights into user engagement, with peaks and valleys indicating periods of high or low activity, prompting analysis of influencing factors as shown in Figure 18.

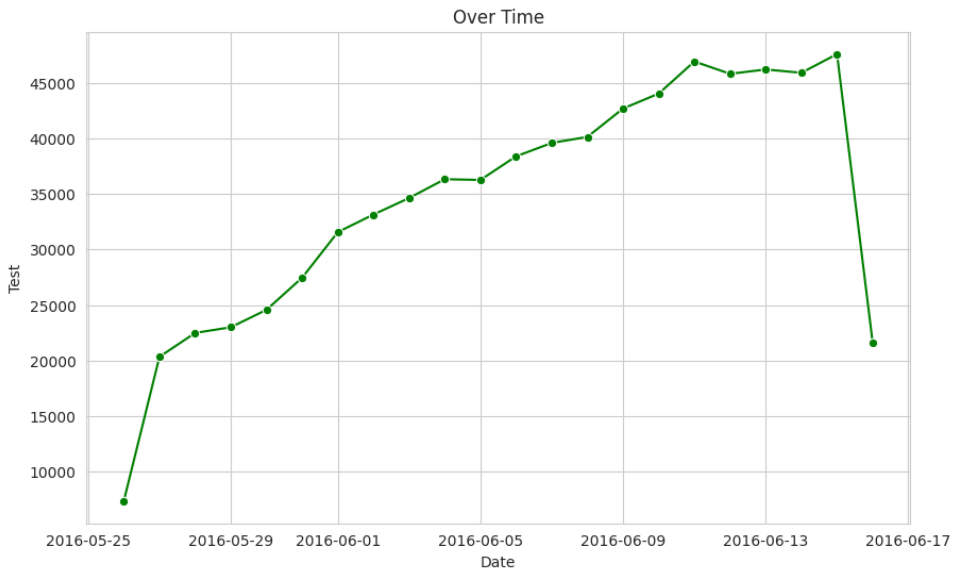


Figure 18: Daily click counts in the game, depicted over time.

7.4 Level Events

7.4.1 Are there any missing values?

There are no missing values present in the dataset as shown in Figure 19.

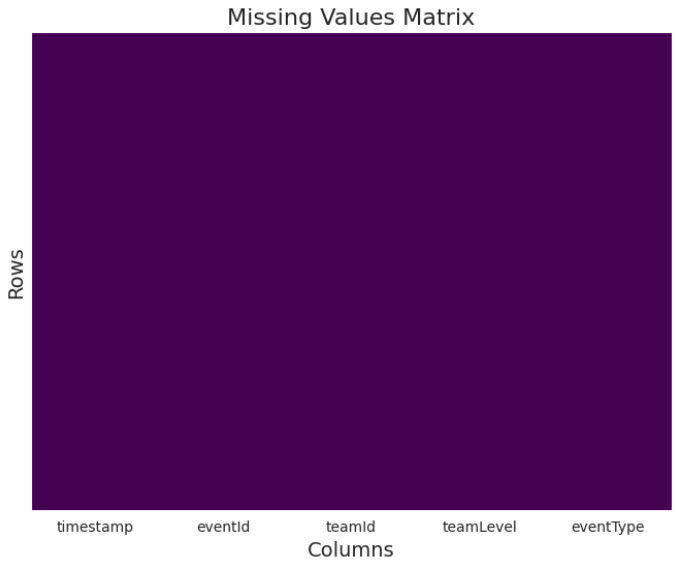


Figure 19: Missing Values in Level Events Dataset

7.4.2 Daily Level Events Over Time

The time series plot depicts daily counts of level events (start or end) in the game, revealing trends over time. It offers insights into user progression and activity, with peaks and valleys indicating periods of increased or decreased engagement in starting or completing levels as shown in Figure 20.

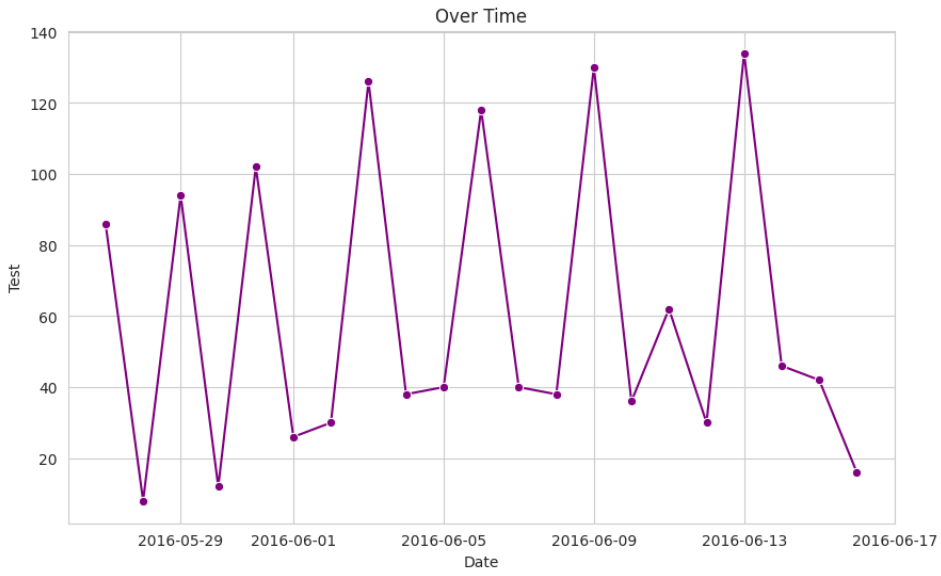


Figure 20: Daily-level events depicted over time in the game.

7.5 Team Assignment

7.5.1 Are there any missing values?

There are no missing values present in the dataset as shown in Figure 21.

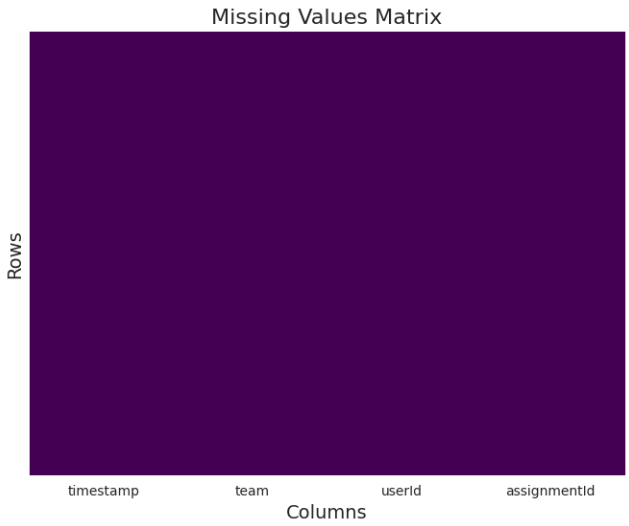


Figure 21: Missing Values in Team Assignments Dataset

7.5.2 Daily Team Assignments Over Time

The time series plot illustrates daily team assignment counts in the game, indicating trends over time. It offers insights into user engagement with team-related activities, with peaks and valleys suggesting fluctuations in joining or leaving teams, prompting further trend analysis as shown in Figure 22.

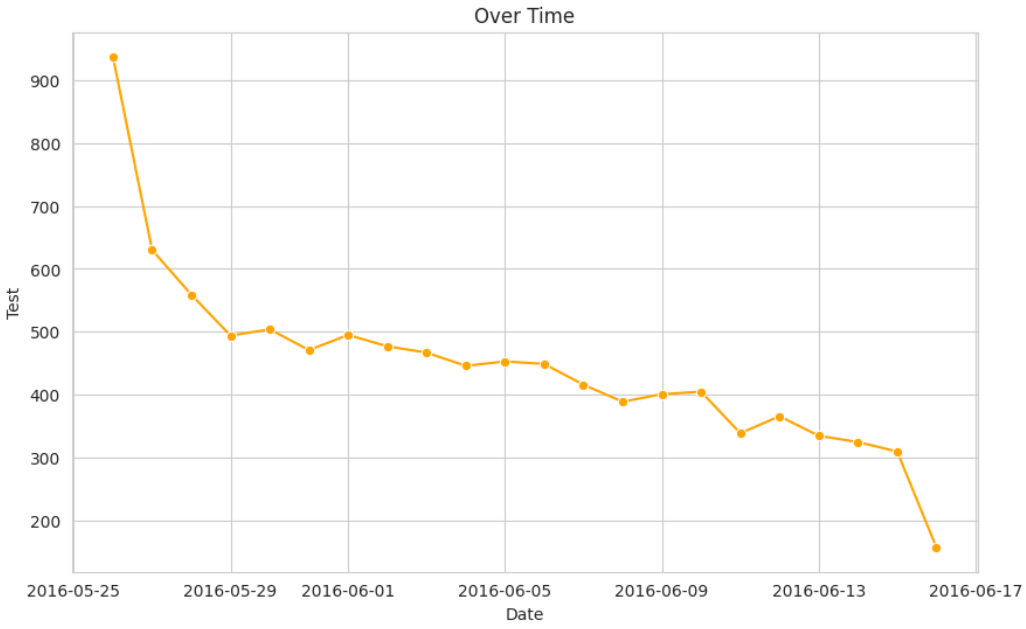


Figure 22: Daily team assignments depicted over time in the game

7.6 Team

4.6.1 Are there any missing values?

There are no missing values present in the dataset as shown in Figure 23.

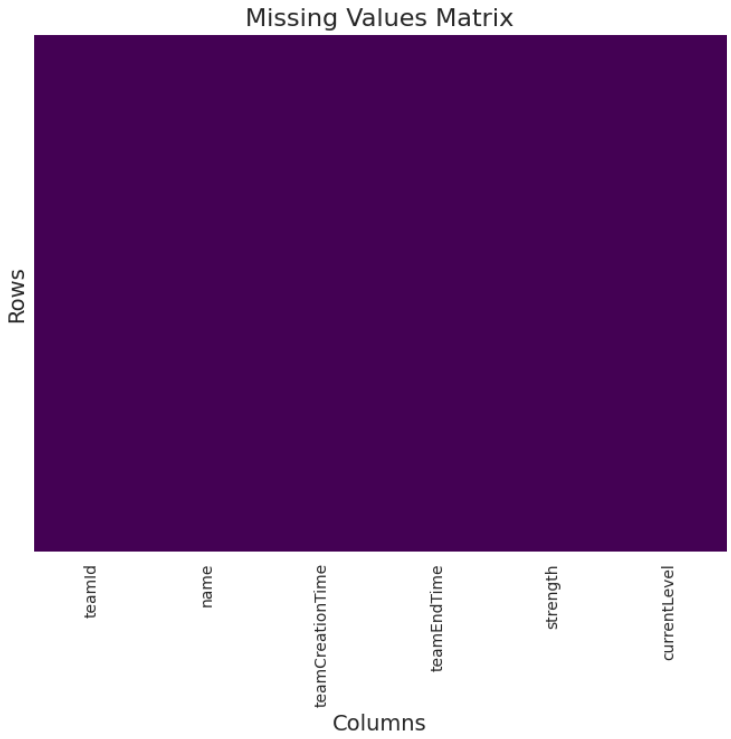


Figure 23: Missing Values in Team Dataset

4.6.2 Daily Team Creation Over Time

The time series plot displays daily team creation counts in the game, revealing trends over time. It provides insights into user engagement with team-related features, with peaks and valleys indicating fluctuations in team creation activity, prompting further analysis of trends as shown in Figure 24.

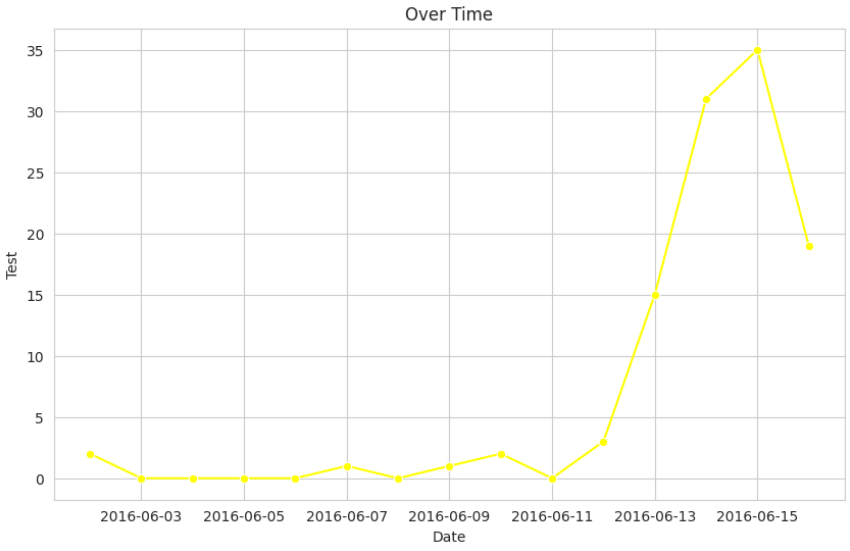


Figure 24: Daily Team Creation Over Time

4.6.3 Top 20 Teams by Strength

It's intriguing that some of the most powerful teams aren't listed among the largest teams. What's particularly interesting is team 9; they rank third in strength and spending. They seem to spend the most while also being one of the strongest teams, despite being the fourth largest in terms of members. On the other hand, teams with lower strength aren't among the biggest. However, the fourth weakest team appears to be spending less compared to others in the lower section as shown in Figure 25.

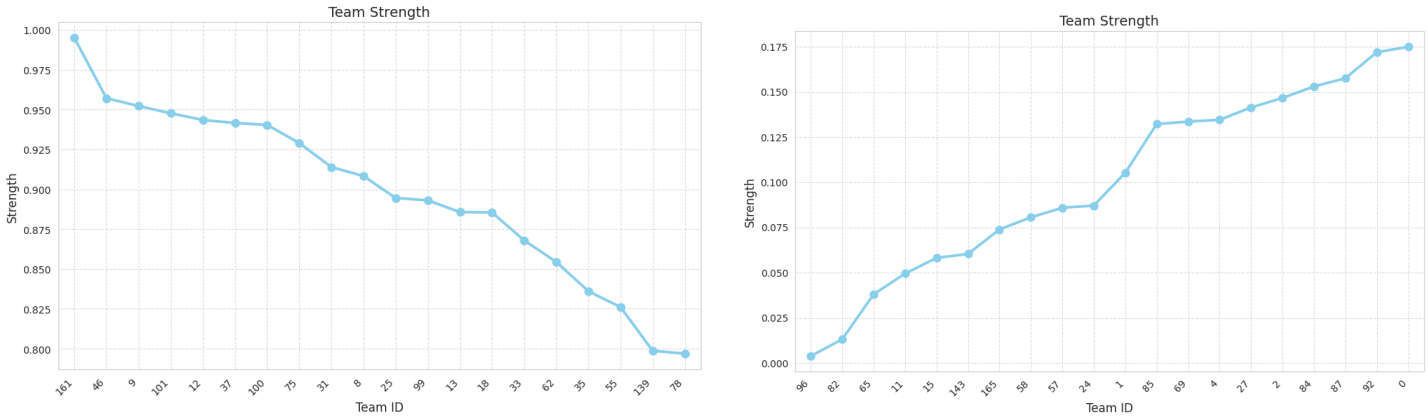


Figure 25: Team Strengths

7.7.1 Are there any missing values?

There are no missing values present in the dataset as shown in Figure 26.

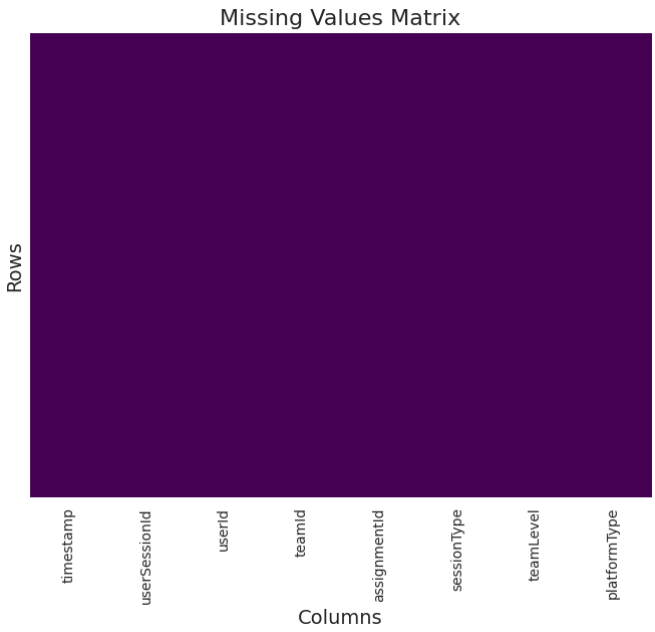


Figure 26: Missing Values in User Session Dataset

7.7.2 Daily User Session Trends

The plot displays daily fluctuations in user sessions, indicating varying levels of engagement over time. Patterns suggest potential seasonal trends and long-term popularity. Anomalies may signify external influences on user activity. Insights guide strategies for enhancing player engagement and retention as shown in Figure 27.

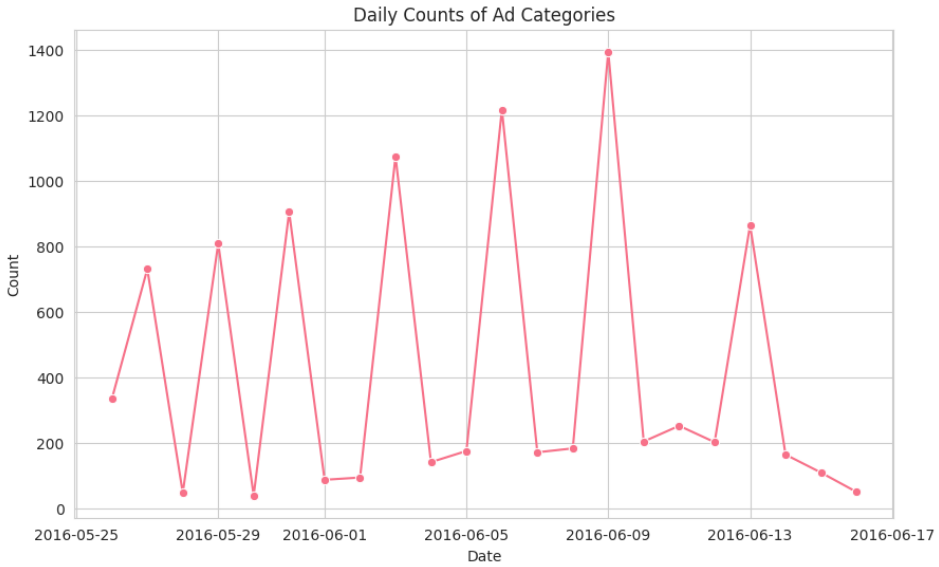


Figure 27: Daily User Session Over Time

7.7.3 Distribution of Session Types by Platform

The stacked histogram shows session type frequency across different platforms. Mac users have the highest session starts or ends, indicating a significant user base. This insight can inform development and marketing strategies targeting Mac users as shown in Figure 28.

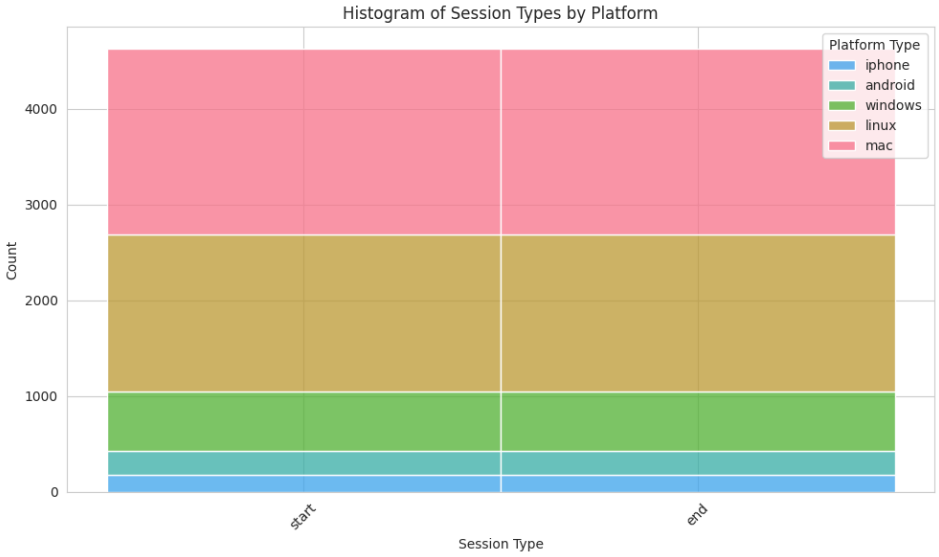


Figure 28: Session Types by Platform

7.8 Users

7.8.1 Are there any missing values?

The initial data frame along this path contains some missing data specifically in the "country" column, and it's only a minor portion as shown in Figure 29.

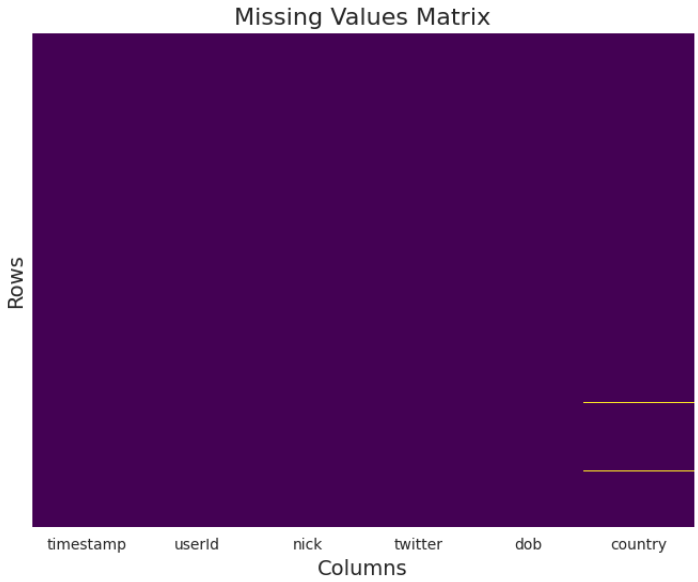


Figure 29: Missing Values in Users Dataset

7.8.2 Distribution of Player Birth Years

The histogram of birth years reveals insights into the distribution of player ages in the game community. One notable observation is the high frequency of birth year 1990, indicating a significant proportion of players born in that particular year as shown in Figure 30.

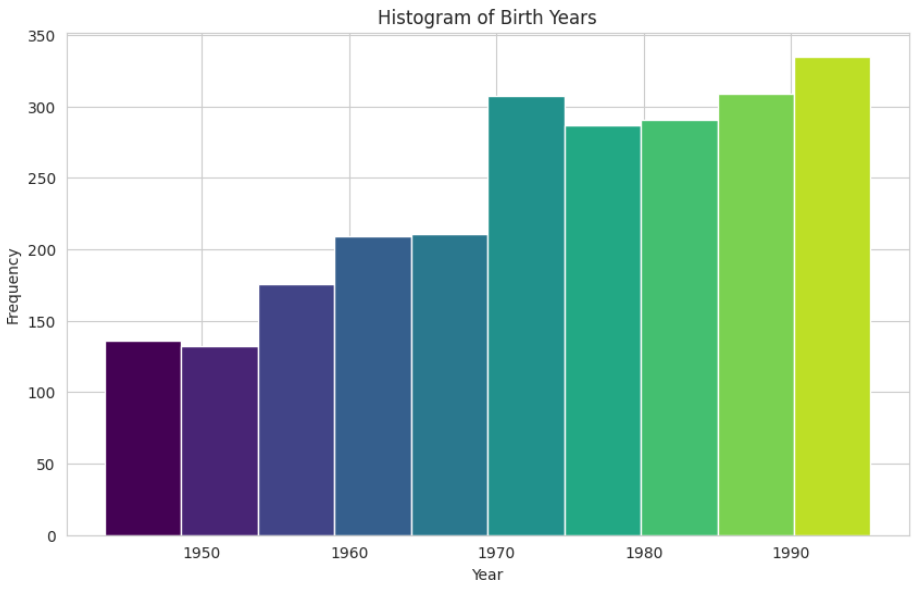


Figure 30: Distribution of Player Birth Years

7.9 Combined Data

7.9.1 Are there any missing values?

The initial data frame along this path contains some missing data specifically in the "count_buyid" and “avg_price” columns as shown in Figure 31.

Figure 31: Missing Values in Combined Dataset

7.9.2 Players' Preferred Devices?

The pie chart displays platform distribution in the dataset, with iPhone at 41.9% and Android at 35.4%. This suggests a preference for iOS among players, guiding developers to optimize the game for iPhone users and tailor marketing strategies accordingly as shown in Figure 32.

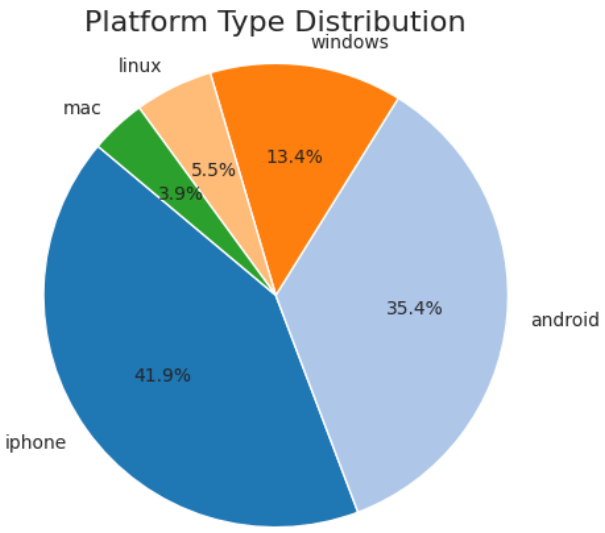


Figure 32: Platform Usage Distribution

7.9.3 Game Clicks vs. Hits on Different Platforms

Linux users show higher hit counts compared to other platforms, suggesting stronger engagement with specific game features. Developers can tailor elements to better suit Linux users, potentially attracting a larger audience from this platform as shown in Figure 33.

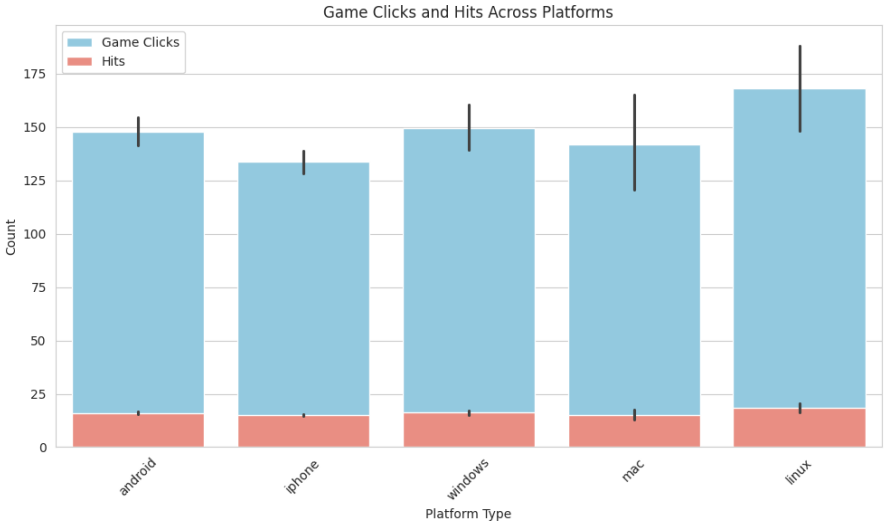


Figure 33: Comparative Analysis of Game Clicks and Hits Across Platforms

7.9.4 Analysing Average Price Distribution Across Platform Types

Histograms for different platforms reveal varying average price distributions. This insight aids in understanding pricing strategies and consumer preferences, guiding decisions for developers and marketers as shown in Figure 34.

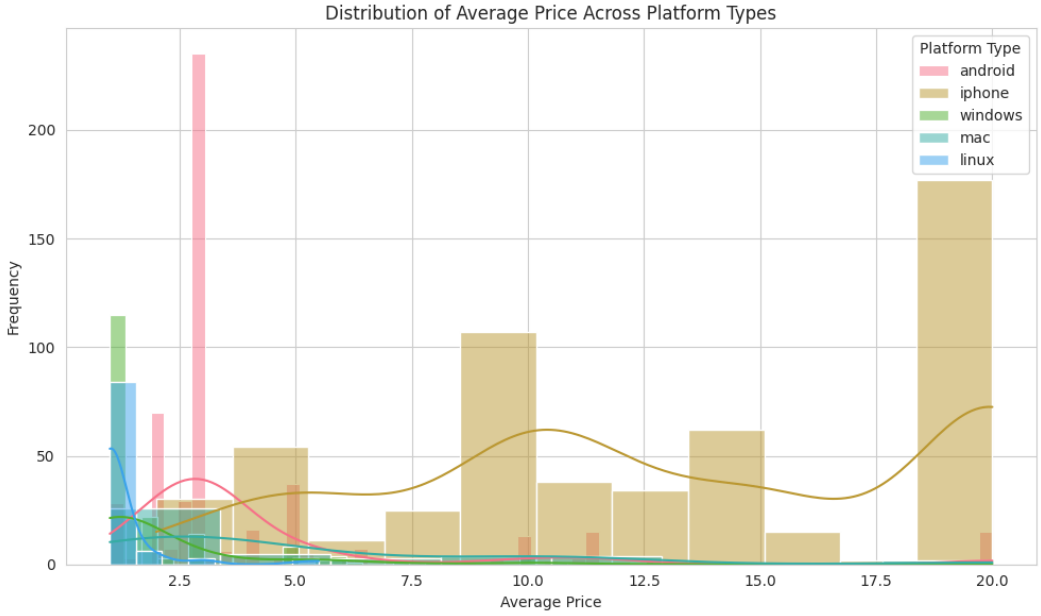


Figure 34: Distribution of Average Prices

7.9.5 Comparative Analysis of Team Levels Across Platform Types

The iPhone platform shows the highest bar for team level 6, indicating a significant player presence at this level. This suggests a correlation between platform type and player progression, influencing game design and marketing strategies targeting iPhone users at higher team levels as shown in Figure 35.

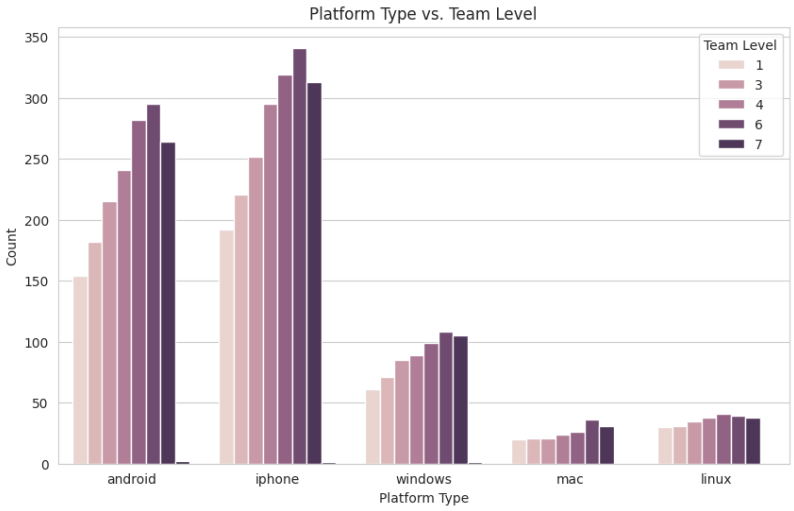


Figure 35: Distribution of Team Levels by Platform Type

7.9.6 Analyzing Conversion Rates Across Platform Types

Windows platform shows the highest conversion rate, indicating greater likelihood of in-game purchases. Developers and marketers can optimize game experience and strategies to enhance engagement and revenue, particularly targeting Windows users as shown in Figure 36.

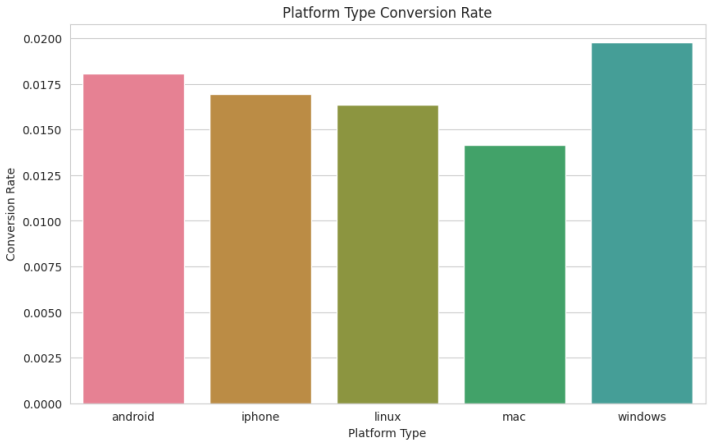


Figure 36: Comparison of Conversion Rates

8. Machine Learning Modelling

8.1 Classification

Classification is like teaching a computer to recognize things. You show it examples (training data) and tell it what they are. Then, you test it to see if it can figure out new examples correctly (test data). Once it learns well, you can use it to identify things it hasn't seen before (Kotsiantis, 2007).

8.1.1 Decision Tree

A decision tree is a flowchart aiding in decision-making, illustrating choices and potential outcomes based on factors like probabilities and costs. It employs if-then rules to determine categories or values, structured like a tree with a root, branches, and leaves (Song, 2015).

The decision tree model evaluation report demonstrates perfect performance with accuracy, precision, recall, and an F1-score of 1.0 for each class. It indicates accurate classification and robust predictive capability, supported by well-distributed instances across classes in test data as shown in Figure 37 and Table 4.

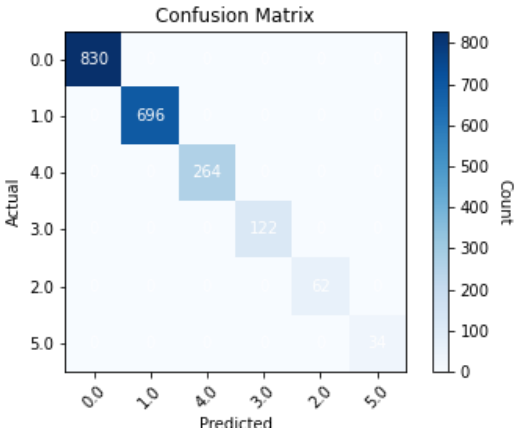


Figure 37: Decision Tree Confusion Matrix

	accuracy	precision	recall	f1-score	support
0.0	1.0	1.0	1.0	1.0	830
1.0	1.0	1.0	1.0	1.0	696
4.0	1.0	1.0	1.0	1.0	62
3.0	1.0	1.0	1.0	1.0	122
2.0	1.0	1.0	1.0	1.0	264
5.0	1.0	1.0	1.0	1.0	34

Table 4: Decision Tree Classification Report

8.1.2 Support Vector Machine

An SVM is like a smart tool that learns from examples. You give it lots of examples of different things, and it learns how to tell them apart. Once it learns, you can show it new things, and it can tell you which category it belongs to (Cervantes, 2020).

The SVM classification report shows moderate performance across classes with accuracy, precision, recall, and F1-score around 0.58. Similar scores across metrics and classes suggest limited classification ability. Insights from the support section highlight potential data imbalance effects as shown in Figure 38 and Table 5.

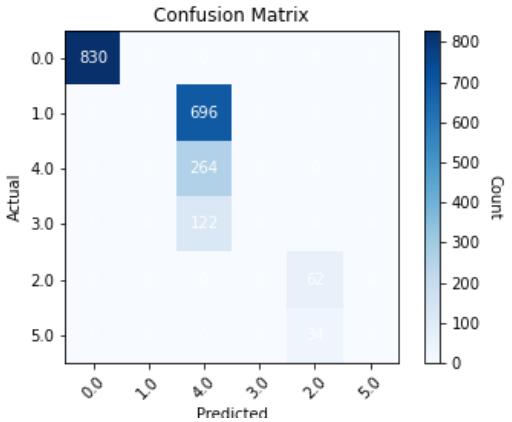


Figure 38: SVM Confusion Matrix

	accuracy	precision	recall	f1-score	support
0.0	0.575697	0.465366	0.575697	0.465366	830
1.0	0.575697	0.465366	0.575697	0.465366	696
4.0	0.575697	0.465366	0.575697	0.465366	62
3.0	0.575697	0.465366	0.575697	0.465366	122
2.0	0.575697	0.465366	0.575697	0.465366	264
5.0	0.575697	0.465366	0.575697	0.465366	34

Table 5: SVM Classification Report

8.2 Clustering

Clustering organizes data points based on similarity without predefined labels. It's unsupervised learning, focusing on finding patterns and grouping similar points together. Methods like distance measurement determine group membership, aiming to create clusters where points are more alike (Ezugwu, 2022).

8.2.1 K-Means

K-means clustering groups data based on proximity, starting with random cluster centers and iteratively refining them. It requires a predefined cluster count and works well with clearly separated data but struggles with overlap, providing no clear evaluation and being sensitive to noise and initialization (Sinaga, 2020).

The clustering performance seems to improve with fewer clusters (k). Notably, the Silhouette Score peaks at approximately 0.95 for cluster numbers ranging from 10 to 20. However, as the number of clusters increases, the Silhouette Score declines, indicating diminishing clustering quality. Hence, opting for a lower number of clusters might be more suitable for this dataset as shown in Figure 39.

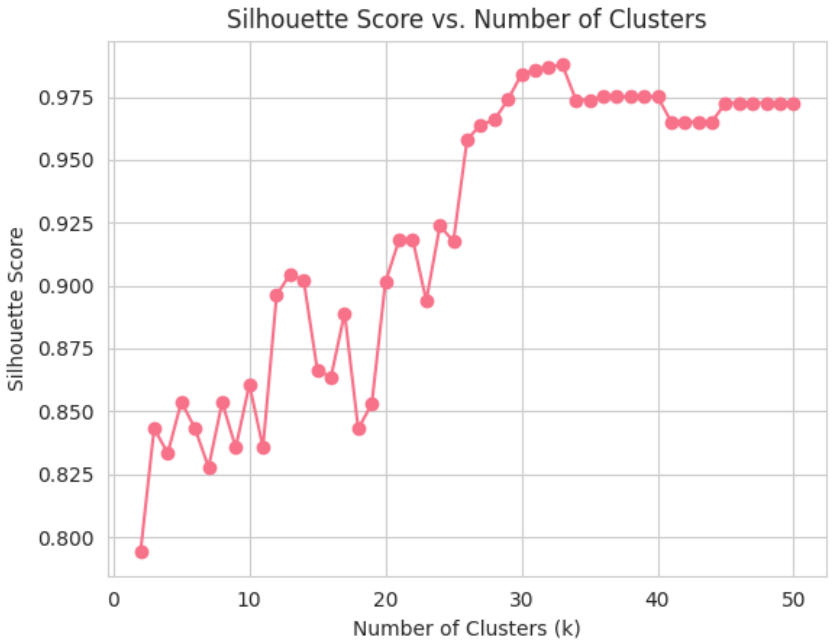


Figure 39: K-Means's Silhouette score vs number of clusters

8.2.2 Gaussian Mixture Models (GMMs)

A Gaussian Mixture Model (GMM) clusters data points or estimates data density by assuming a mix of Gaussian distributions. It assigns likelihoods to each point belonging to each cluster, accommodating potential membership in multiple clusters simultaneously. GMM is widely applied in machine learning for pattern recognition (Wang, 2019).

The "Silhouette Score vs. Number of Clusters" graph assesses clustering effectiveness, showing higher silhouette scores indicate better matches within clusters. Variability increases with more clusters, suggesting potential overfitting. Optimal cluster selection involves balancing performance and stability as shown in Figure 40.

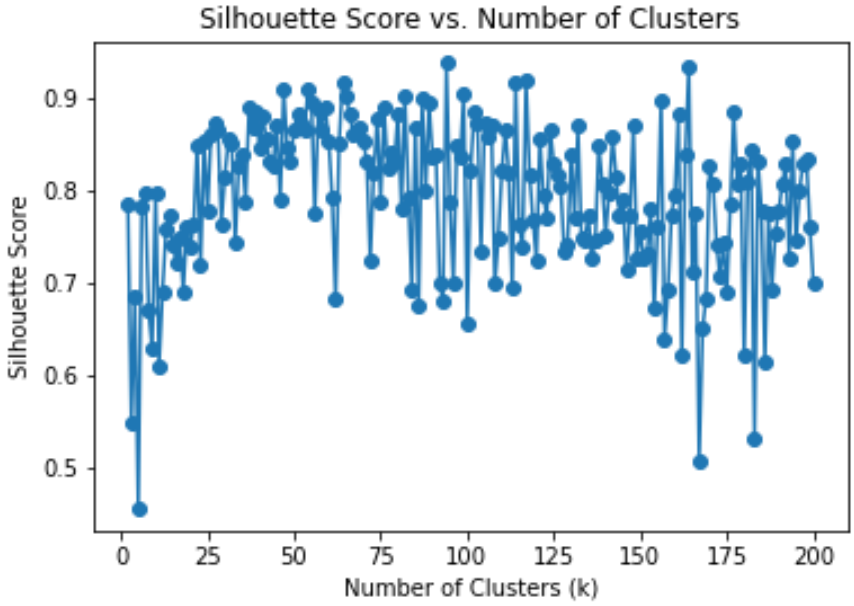


Figure 40: GMM's Silhouette score vs number of clusters

9. Graph Analysis

9.1 Chat Items Created in Team Chat Sessions

The graph represents interactions in team chat sessions, where users create chat items. Nodes represent users, chat items, and team chat sessions, while relationships indicate who creates which chat item and which chat item is part of which team chat session as shown in Figure 41.

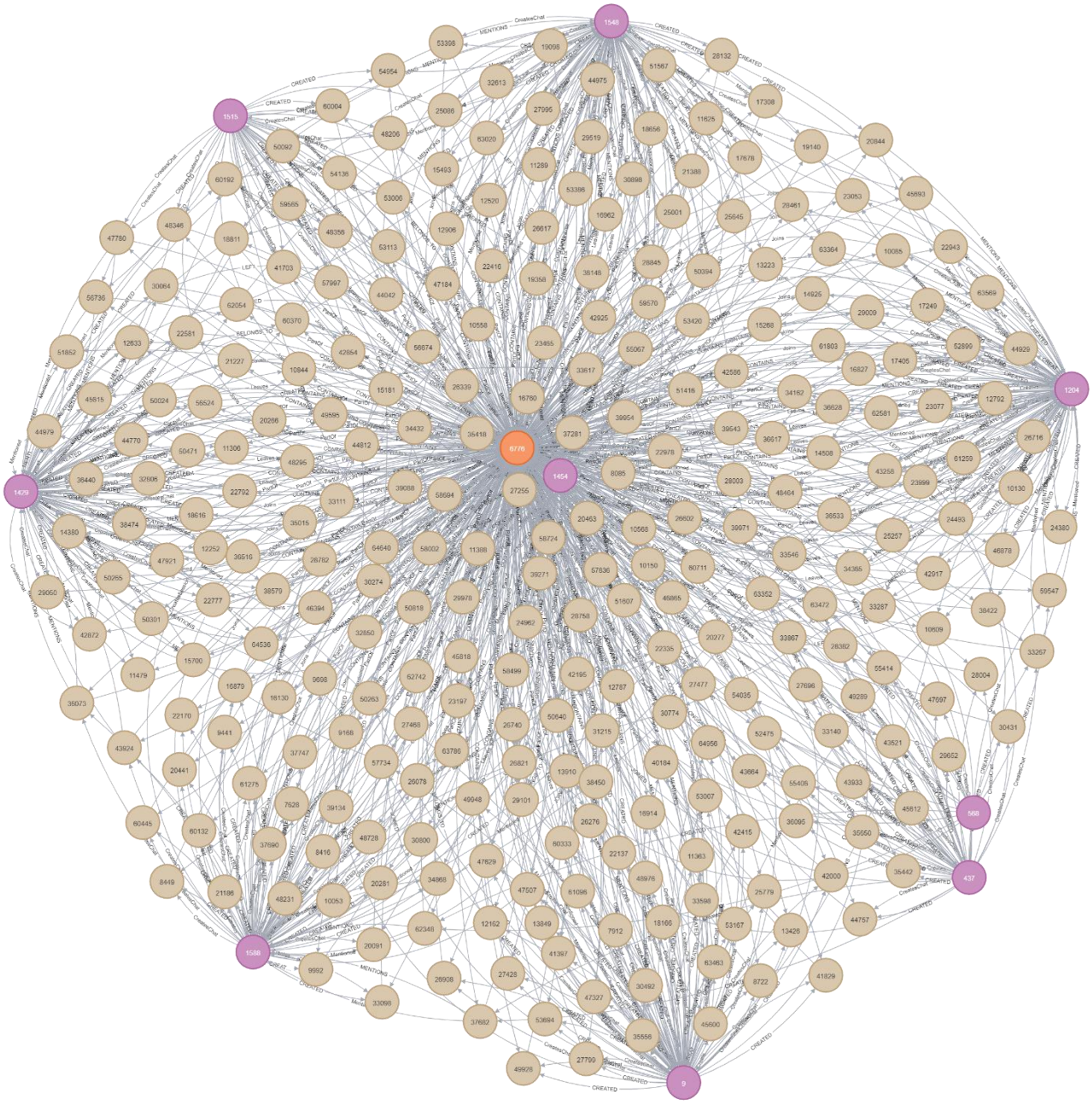


Figure 41: User-Team Chat Interactions

9.2 Team Chat Sessions Created by Users

The graph visualizes the creation of team chat sessions by users within their respective teams. Nodes represent users, team chat sessions, and teams, while relationships indicate who creates which session and which session is owned by which team as shown in Figure 42.

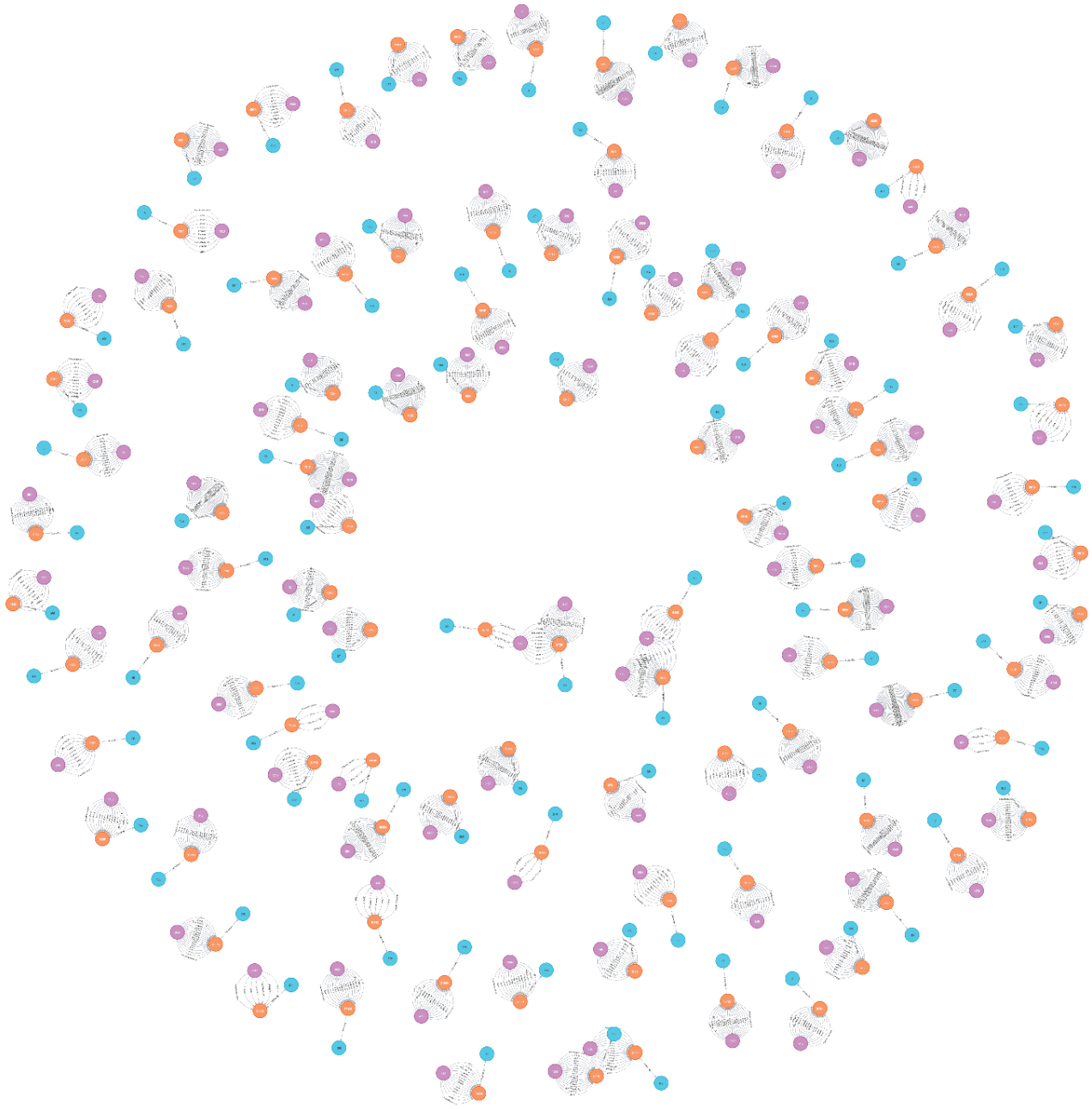


Figure 42: Creation of team chat sessions by users within their teams.

9.3 Users Joining Team Chat Sessions

The graph displays users joining team chat sessions over time. Nodes represent users and team chat sessions, while relationships indicate users joining specific sessions at given timestamps as shown in Figure 43.

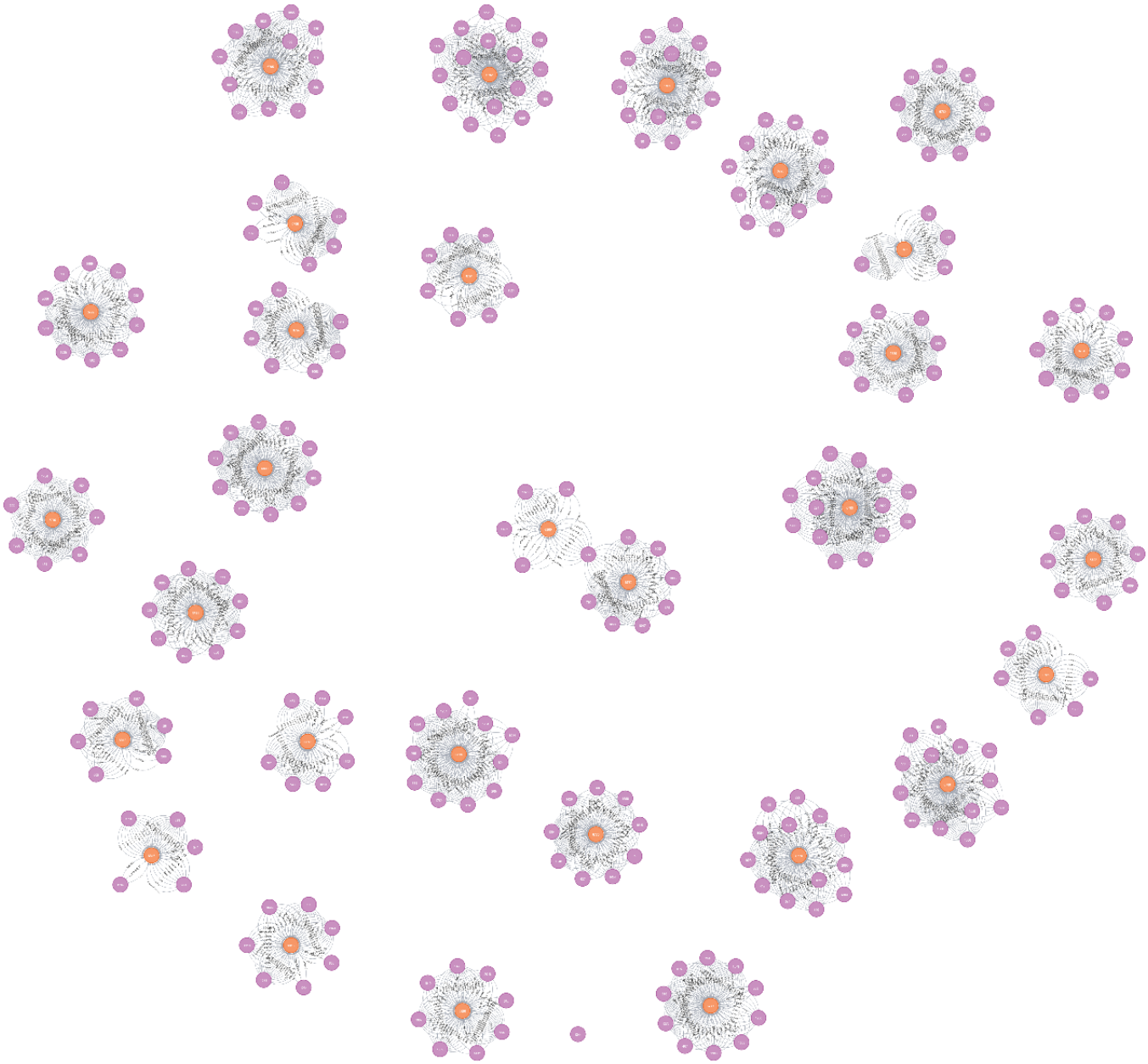


Figure 43: Users joining team chat sessions

9.4 Users Leaving Team Chat Sessions

The graph illustrates users leaving team chat sessions with timestamps. Nodes represent users and team chat sessions, while relationships indicate users leaving specific sessions at given timestamps as shown in Figure 44.

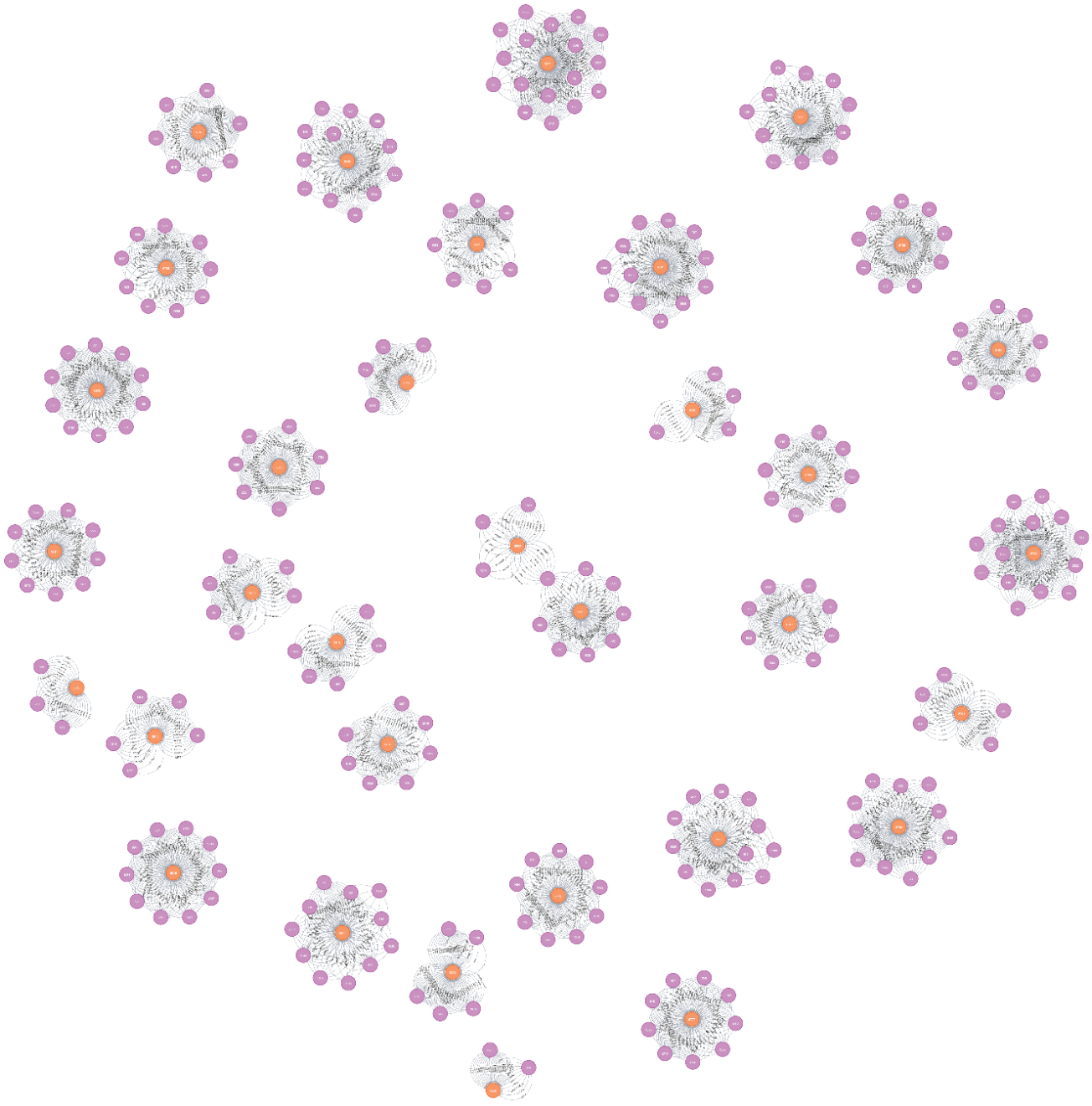


Figure 44: Users leaving team chat sessions

9.5 User Mentions in Team Chat

The graph depicts instances where users are mentioned in team chat items, along with timestamps. Nodes represent chat items and users, while relationships indicate mentions with timestamps as shown in Figure 45.

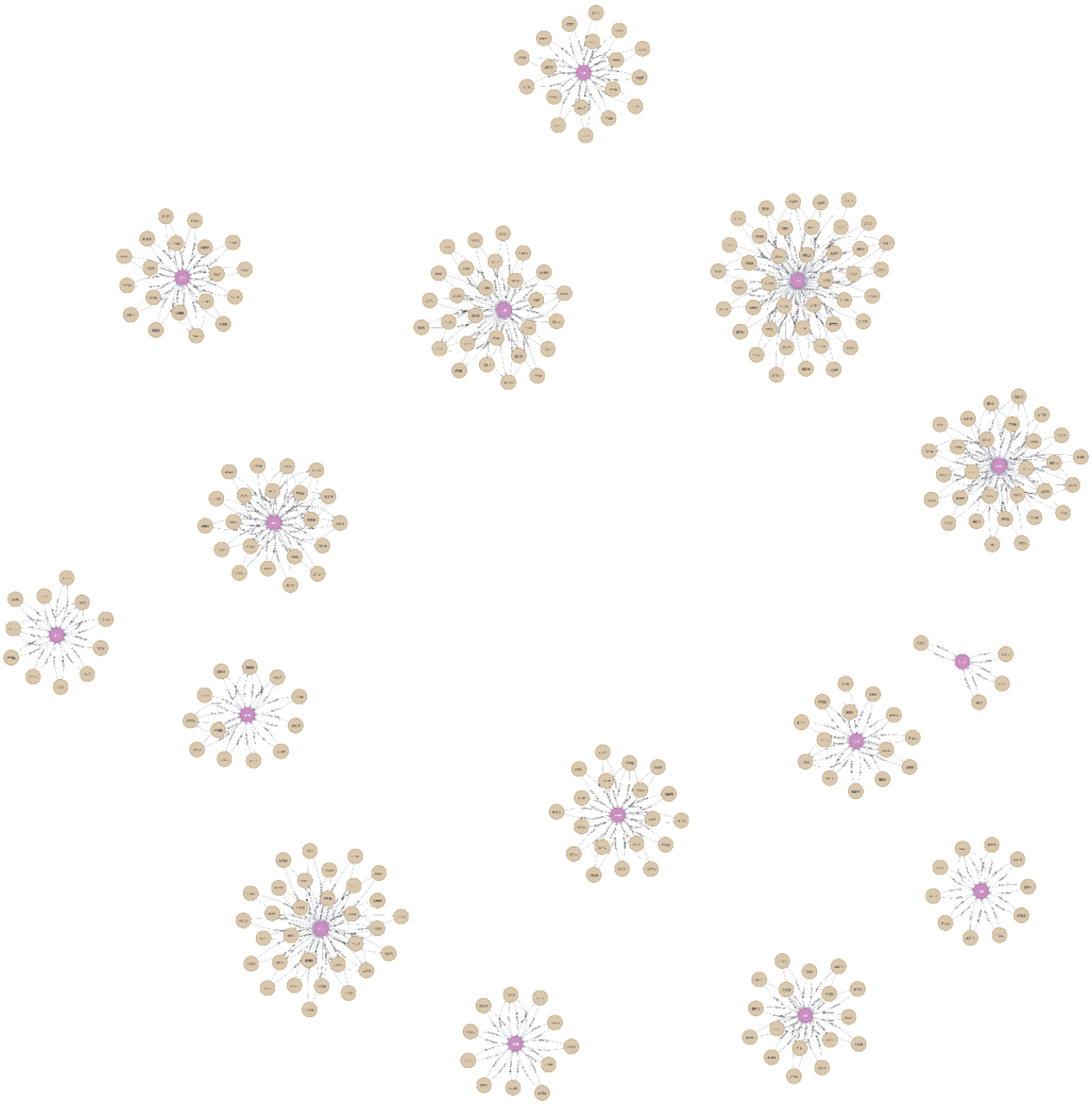


Figure 45: User mentions in team chat items

9.6 Responses in Team Chat

The graph illustrates responses made by users to chat items in team chat, along with timestamps. Nodes represent chat items and users, while relationships indicate responses with timestamps as shown in Figure 46.

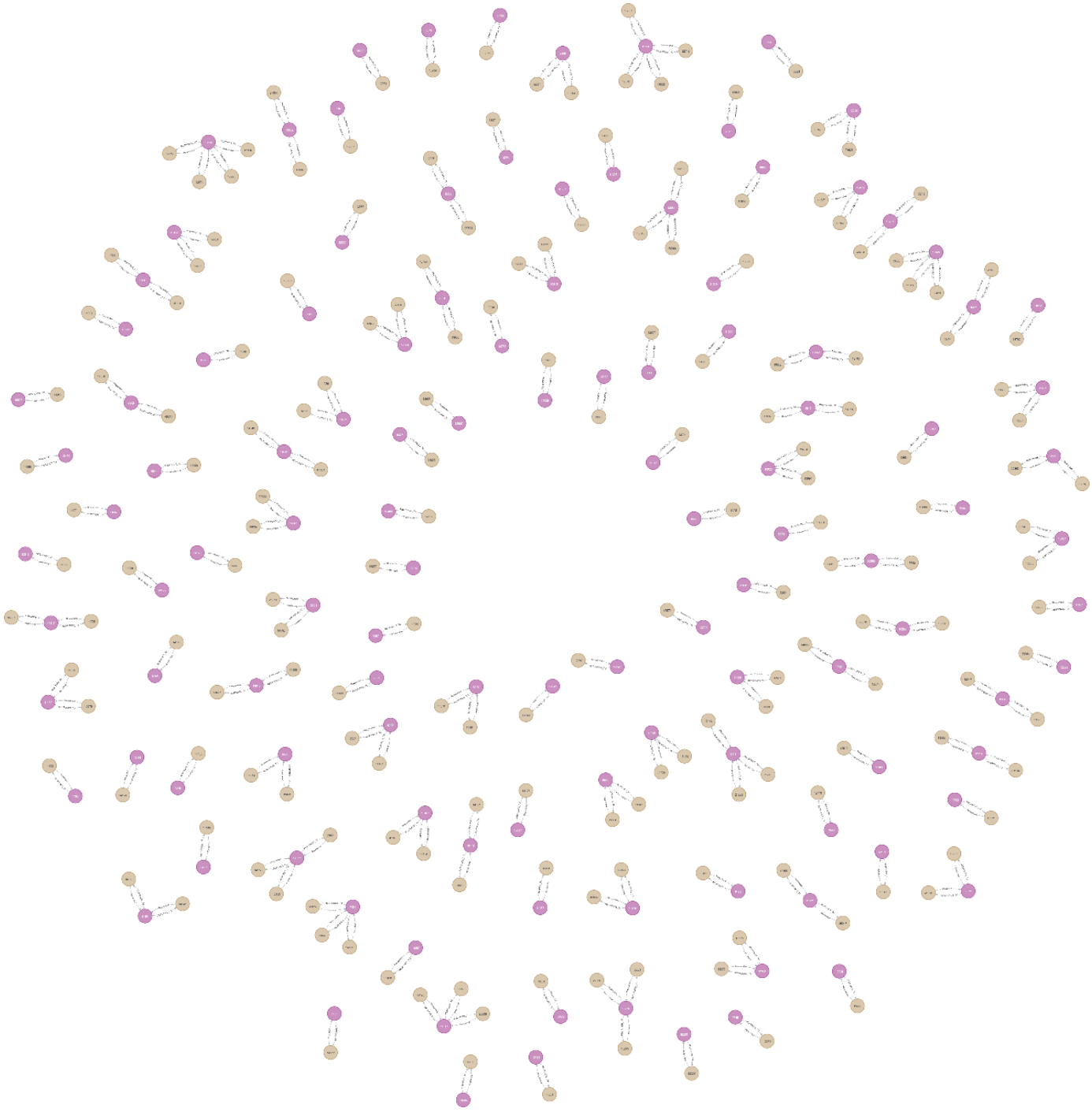


Figure 46: Responses made by users to chat items in team chat

10.Role of Ethics

Aspect	Description	Applicable?
Data Quality	Ensuring data accuracy, completeness, and consistency is crucial for ethical data storage and processing. Ethical considerations include responsible interpretation, ongoing policy initiatives, and environmental impact.	Yes
Machine Learning	Ethical considerations in machine learning involve transparency, accountability, fairness, and bias mitigation. Responsible interpretation and management, as well as ongoing policy initiatives, are essential for addressing ethical concerns in machine learning algorithms.	Yes

Table 6: Role of Ethics

11.Conclusion

"Catch The Pink Flamingo" showcases the efficacy of Big Data analytics. Through rigorous Exploratory Data Analysis (EDA), diverse machine learning models for classification and clustering, and innovative graph analytics with Neo4j, we uncovered valuable insights and patterns. EDA provided a solid foundation, while machine learning models revealed nuanced understandings of the data and its structures. Neo4j's graph analytics unearthed hidden relationships and communities. Together, these approaches offer a comprehensive understanding of the dataset, paving the way for actionable intelligence. Our project exemplifies the transformative potential of leveraging advanced analytics techniques in Big Data ecosystems, promising continued exploration and discovery in data-driven endeavors.

12.Limitations and Recommendations

Despite the successes of "Catch The Pink Flamingo," several limitations were encountered throughout the project, including challenges related to data quality and conversion issues, which merit consideration for future endeavors.

Firstly, Data quality profoundly affects analysis. Our project faced conversion problems due to inconsistent formats, missing values, and encoding differences, leading to noise and biased results. Robust quality assurance and preprocessing are crucial for reliable, representative datasets, mitigating such challenges for accurate analysis and modeling.

Secondly, Algorithm selection impacts model performance; data conversion issues may hinder evaluation. Future work may focus on exhaustive model search or novel techniques addressing conversion challenges.

Additionally, the scalability and computational requirements of graph analytics can pose challenges, particularly with larger datasets. Optimizing algorithms and leveraging

distributed computing frameworks may help alleviate these constraints, enabling more efficient analysis of complex graph structures.

In light of these limitations, several recommendations can be made to enhance future projects in similar domains. Invest in data quality assurance and preprocessing, experiment with diverse algorithms, utilize scalable graph analytics methods, and prioritize interpretability and transparency in model development and reporting.

13.Source Code

GitHub Link: https://shorturl.at/cjxLZ

14.References

Casado, R. &. Y. M., 2015. Emerging trends and technologies in big data processing. Concurrency and Computation: Practice and Experience. Volume 27(8), pp. 2078-2091.

Cervantes, J. G.-L. F. R.-M. L. &. L. A., 2020. A comprehensive survey on support vector machine classification: Applications, challenges and trends.. Volume 408, pp. 189-215.

Ezugwu, A. E. I. A. M. O. O. O. A. L. A. J. O. E. C. I. &. A. A. A., 2022. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges,.

Gogtay, N. J. &. T. U. M., 2017. Principles of correlation analysis.. *Journal of the Association of Physicians of India,* pp. 78-81.

Good, I. J., 1983. The philosophy of exploratory data analysis. pp. 283-295.

Kotsiantis, S. B. Z. I. &. P. P., 2007. Supervised machine learning: A review of classification techniques.. Volume 160(1), pp. 3-24.

Li, Z. L. Z. X. K. &. L. X., 2023. Evaluating LLM's Code Reading Abilities in Big Data Contexts using Metamorphic Testing.. *9th International Conference on Big Data and Information Analytics (BigDIA),* pp. 232-239.

Ma, S. W. H. M. L. W. L. W. W. H. S. .. &. W. F., 2024. The Era of 1-bit LLMs: All Large Language Models are in 1.58 Bits. pp. 2402-17764.

Saeed, N. &. H. L., 2021. Big data characteristics (V's) in industry. *Iraqi Journal of Industrial Research,* Volume 8, pp. 1-9.

Sinaga, K. P. &. Y. M. S., 2020. Unsupervised K-means clustering algorithm.. *IEEE.*

Song, Y. Y. &. Y. L. U., 2015. Decision tree methods: applications for classification and prediction.. *Shanghai archives of psychiatry,* Volume 27(2), p. 130.

Wang, Z. D. C. C. R. M. &. F. B., 2019. Comparison of K-means and GMM methods for contextual clustering in HSM.. pp. 154-159.

Zhou, Y. G. C. W. X. C. Y. &. W. Y., 2024. A Survey on Data Augmentation in Large Model Era.

Zou, X., 2023 . New Opportunities for AI Innovation with Big Data: Indirect Docking between GLPS and LLM. *6th International Conference on Artificial Intelligence and Big Data (ICAIBD), IEEE,* pp. 444-450.