



CIS 8395 Big Data Analytics Experience Project
(Spring 2024)

Final Paper



IPLGPT

WHERE AUTHENTIC TEA MEETS AI-
POWERED
IPL CHATBOT

Team Name- CHAIGPT

Dhananjay Ghate
Neena Chaudhari
Purvi Bharani
Shravan Honade

Table of Contents

1.	Business Problem.....	3
2.	Proposed Solution	5
3.	Real World Application.....	6
3.1	Fans	6
3.2	Advertisers & Sponsors	7
3.3	Influencers & Bloggers	7
4.	Data Strategy.....	8
4.1	Data Sourcing.....	8
4.2	Data Frequency	10
4.3	Data Validation	10
5.	Workflow	11
5.1	Data Orchestration	11
5.2	IPLGPT Orchestration Workflow.....	13
6.	ETL (Extraction, Transformation, and Load)	14
6.1	Extraction	14
6.2	Transformation	16
6.3	Load.....	19
7.	Exploratory Data Analysis.....	21
7.1	Tableau.....	21
8.	LLM	24
9.	RAG Architecture	25
9.1	Introducing RAG.....	25
9.2	Tools Used:	26
10.	Streamlit.....	29
11.	AI Agents	33
11.1	What are AI Agents?.....	33
11.2	Our AI Agent - IPLGuru	35
12.	My Conclusion.....	Error! Bookmark not defined.
13.	Reference	42

1. Business Problem

Imagine being a marketing executive for one of the big sports organizations in India that are all getting ready for the IPL next season. You have a problem: fan engagement and revenue are completely integrated with advertising and there is high competition to promote yourself to fans before an IPL. The picture is crystal clear: you want to bring fans closer to the IPL, take passive viewers, and turn them into active partisans, dancing zombies supporting a multi-billion-dollar brand. These are noble objectives for an organization. But there's only so much that clever marketing can achieve. Beyond its stellar performances on the field, there must be a suite of platforms and experiences that provide fans with real-time engagement opportunities. One could start by going beyond traditional match broadcasts and building a suite of platforms that appeal to key demographics. It is not difficult to imagine game-day updates, detailed match summaries, in-depth analyses or quizzes, and trivia sections that keep the fans glued and provide the conceptual backdrop for some of the monetization strategies that enable the existence of the whole platform.

But engagement on its own is not enough. To make an effect, you need to foster a sense of network amongst lovers, developing a space where they can connect, percentage their ardor, and feel like they may be part of something bigger than themselves. By facilitating discussions, encouraging interactions, and supplying a platform for fan-generated content, you may strengthen the bond between fanatics and the IPL logo, building loyalty and advocacy inside the method.

Of direction, no engagement approach would be entirely without considering the position of fantasy sports. Fantasy cricket has exploded in reputation in recent years, supplying lovers with a new way to revel in the fun of the sport. Integrating fable game support into your platform, not only caters to the developing demand for delusion cricket but also deepens fanatics' investment within the match, giving them a stake in the outcome of every in-shape.

And finally, there is the query of logo presence. In a sea of competing voices and distractions, how do you make certain that your emblem sticks out? The secret is to weave your emblem messaging seamlessly into the cloth of the fan experience, imparting price-introduced content material, different offers, and subsidized promotions that enhance as opposed to detracting from

the overall revel. By aligning your brand with the passion and pleasure of the IPL, you could grow logo visibility and build high-quality institutions that resonate with enthusiasts long after the final wicket has fallen.

In essence, the enterprise trouble you face is how to create a holistic fan engagement method that leverages the strength of era, community, and branding to raise the IPL experience and power tangible enterprise outcomes. It's a mission that calls for creativity, innovation, and a deep expertise of what makes enthusiasts tick – however, the rewards are nicely well worth the effort.

2. Proposed Solution

The involvement of IPL GPT in the IPL is something crucial for improving the fan's view during IPL time. It provides recent updates, IPL statistics, and player history, offers personalized content, and helps cricket fans have a fantasy cricket, meeting the requirements of different types of cricket audiences. Through building a community, providing access to intermittent content, and integrating branded content on the platform, the IPL GPT gives the venue for fans to network, interact with, and be a part of the exhilaration of the games. Moreover, it becomes a great platform for brands and advertisers to get a chance to raise awareness and level of engagement, which explains why this tool is significant not just for the fans but for those stakeholders associated with the IPL industry.

2.1 Recent Updates and Statistics:

IPL GPT provides real-time updates on matches, scores, player performances, and team standings. For example, it delivers instant notifications about match results, top scorers, and key highlights.

It compiles comprehensive IPL statistics such as batting averages, bowling figures, team records, and historical data. Fans can access this information to analyze player performances and team strategies.

2.2 Community Building and Interactions:

IPL GPT facilitates community building by providing forums, chatrooms, and social media integrations for fans to connect, share opinions, and discuss match scenarios. For example, it hosts live chats during matches for fans to interact in real-time.

Fans can participate in polls, quizzes, and prediction contests organized by IPL GPT, fostering a sense of belonging and camaraderie among cricket enthusiasts.

3. Real World Application

Let's break down the target audience for the IPLGPT into specific groups:

3.1 Fans

Fans are the number one target market for the IPL GPT. This includes cricket lovers of every age, gender, and demographic who observe the IPL closely and are passionate about the game. They seek actual-time updates, player statistics, IPL history, delusion cricket aid, and a sense of community.



Betting & Gaming Enthusiasts: This team consists of people aged between 20 to 42-years-old, and they enjoy placing bets or playing fantasy sports, especially during IPL matches. Prediction, insights and strategies, which provide the best with a wide array of tools are looked for to help the best in the betting decisions and team selections. IPLGPT has a possibility to create their audience through providing professional analysis, history of player performance, injury updates and other essential info to enrich the gaming experience.

3.2 Advertisers & Sponsors

Advertisers and sponsors are vital entities that contribute towards the overall IPL ecosystem and therefore tap into a market of highly loyal and engaged cricket crowds. They actively seek advertising platforms and products promotion, specifically through the social media channels and paid media content. They use integrated marketing and multi-channel campaigns to get their brands and goods known. The influencer marketing platform GPT of IPL serves advertisers and sponsors with the opportunity to have engagement with their target audience in a meaningful way by applying such data of chatbot such as reach and engagement metrics for company promotion and customer engagement.

3.3 Influencers & Bloggers

Major part of the talk of town and those who control the talk of town which has a big impact on the discussion and opinion of cricket community is the influencers and bloggers. They are avid followers on the various social media platforms and their blogs, and their recommendations based on their reviews and endorsements can have a great effect on fans and brand outlook. IPLGPT will work with influencers and bloggers to widen its reach and create waves, to go viral with the content that will be created thoroughly by diving into the target audience's psychology. This partnership can imply this, guide appearance, exclusive interview, backstage cover up so that the two content and the involvement of IPL leads to a high score collectively.

4. Data Strategy

A data strategy states an entity's long term goals concerning data management and includes the organizational activities of gathering, storing, sharing, and usage of the data. It helps simplify data processing at every stage of your data-steerage journey both for itself and the organization you're in.

4.1 Data Sourcing

In this report, we discuss the process of scraping data from official website of Indian Premier league (IPL) and other site of CricTracker using Python beautiful soup library. Web scraping means gathering data from online sites making use of a computer program.

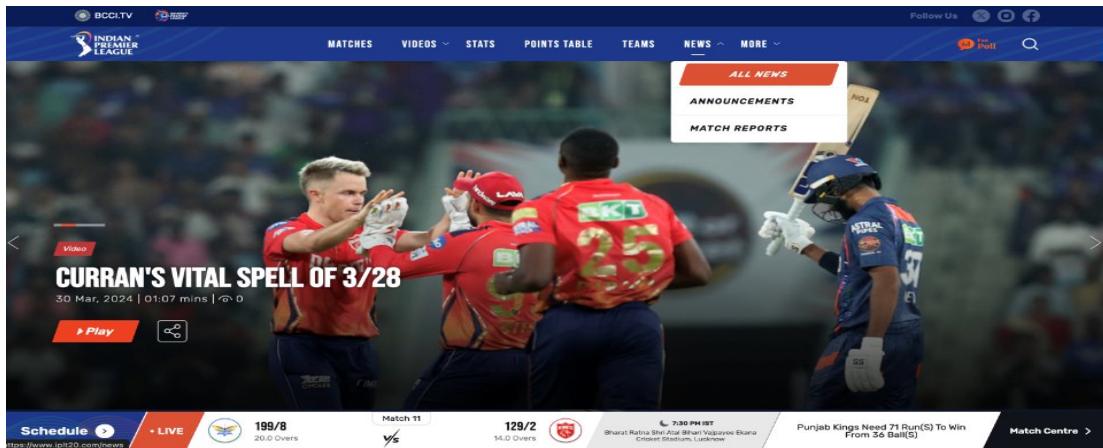
The objective of this data scraping task is to pinpoint specific information like articles, headlines, and stats from IPL official website and CricTracker that will then be weighted and used in IPLGPT.

We followed the methodology described below:



- **Fetching HTML Content:** We utilized the Python requests library to fetch the HTML content of the IPL official website (<https://www.iplt20.com/>) is a platform that aggregates updates, whereas CricTracker (<https://www.crictracker.com/>) is a platform that provides analysis and opinion of a range of cricket related topics. This implied accessing and requesting the information residing on every URL using respective HTTP requests and obtaining the HTML responses containing data of our choice. Since

the data is being fetched from external websites (IPL official website and CricTracker) using HTTP requests, it falls under the category of third-party data.



- **Parsing HTML with Beautiful Soup:** Then, we downloaded the HTML, and the BeautifulSoup library was used to delimit the HTML and recreate it in a structured form. BeautifulSoup offers a powerful source of HTML acquiring operation such as it provides forming structure and architecture to search for specific information.
- **Extracting Data:** By using BeautifulSoup, we found needed and took out it from the HTML structure the parsed. Headlines, the latest news articles and any other update that could be similarly making news on the IPL website and CricTracker were entered there as well.
- **Data Processing:** Following the extraction, we were able to work on the data as was needed. Cleaning data meant removing errors, formatting it correctly among other presenting operations.

In our project, we implemented time-to-time upgrades to the web scraping procedure. Together with the issue resolution mechanisms, we have taken the easy way out to address the potential problems experienced during scraping. The issue of ethics and legality dominates our undertakings. This is why we visibly follow website terms of service and copyright law. These kinds of things protect the recycling data and represent the organization's principle of doing things in a fair, ethical, and legal way.

4.2 Data Frequency

Implementing delta logic in web scraping facilitates a continuous flow of data. This is particularly valuable for websites that update frequently, such as those providing post-match news reports. By employing delta logic, our scraping process ensures that updated information is promptly gathered and integrated into our database, enabling us to stay current with the latest developments.

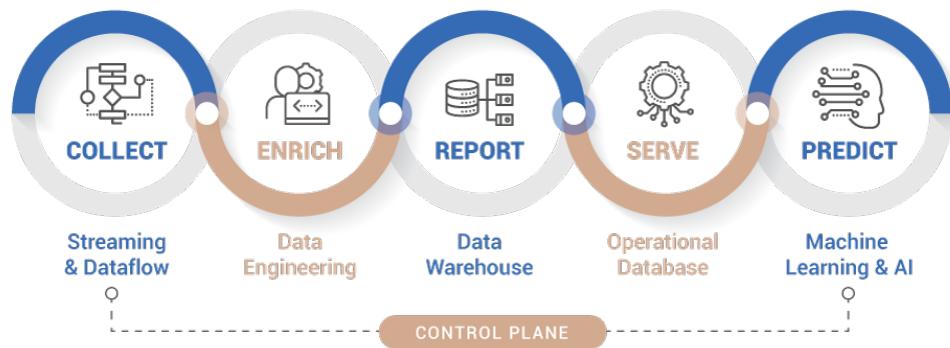
4.3 Data Validation

When it comes to IPL web scraping, data validation plays a pivotal role in maintaining the integrity of the collected data. By implementing robust validation processes, we can ensure that the scraped data accurately reflects the latest updates and developments in the IPL matches. This involves verifying the correctness and completeness of information such as match scores, player statistics, and team standings against reliable sources. Effective data validation not only enhances the quality of our dataset but also ensures that our analysis and insights are based on accurate and reliable information, thus supporting informed decision-making in IPL-related activities.

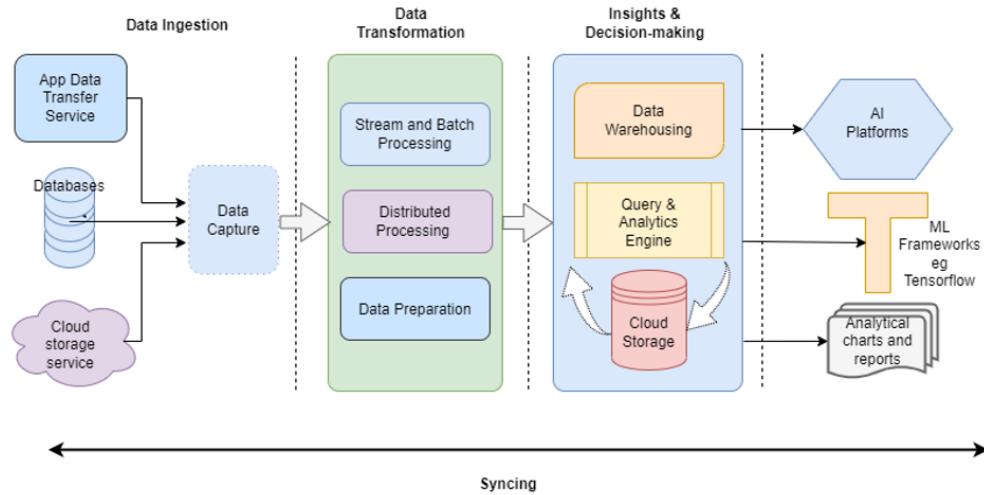
5. Workflow

5.1 Data Orchestration

Orchestration is managing information involves organizing and storing data systematically for easy access and analysis. Correlating data streams entails identifying relationships between different sources to derive meaningful insights. Together, these processes enable IPLGPT to access relevant, structured data and generate informed responses efficiently.



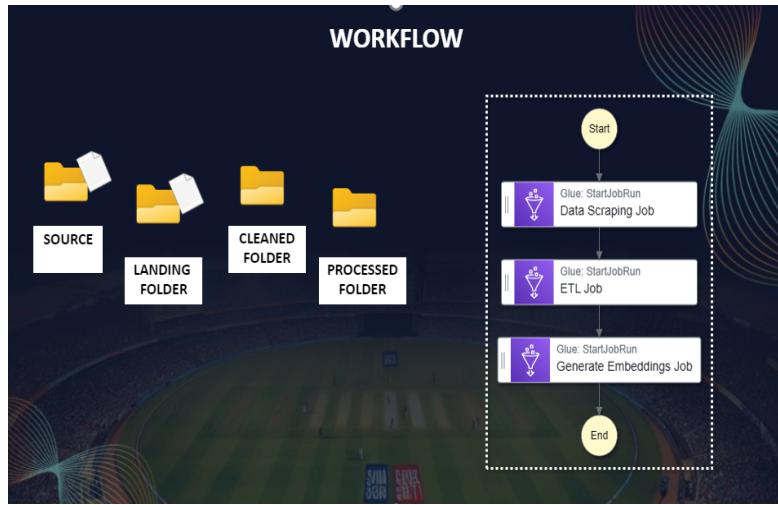
Data orchestration automates the movement of data from its source to storage by consolidating multiple pipeline tasks into a seamless end-to-end process. Its primary objective is to enable data analysis tools to filter, sort, organize, and publish intricate data within a cloud-based data storage system. Additionally, it aids in ensuring compliance with data privacy regulations, eliminating data bottlenecks, and upholding data governance standards.



Common problems in data orchestration include data silos, data quality issues, and connectivity issues, which are widely recognized as troublesome.

- Data silos occur when data becomes segregated across various tools and systems, leading to incomplete insights in one area and distrust of analytics in another. Coordinating data across departments to prevent duplication and establish a single source of truth enables businesses to harness the full benefits of their information.
- Data quality issues arise when combined data contains errors resulting from different teams using disparate naming conventions. Cleaning data becomes a priority when addressing these issues.
- Integration problems stem from the manual connection of various tools and systems for supply and delivery, which can be tedious and difficult to oversee. Automation tools like Twilio Segment offer integration blueprints that significantly reduce the time needed to set up these systems.

5.2 IPLGPT Orchestration Workflow



- **Data Acquisition (Landing Zone):**

A data scraping job retrieves information from a specific website. The extracted data is temporarily stored in a designated "Landing Folder." This folder acts as a staging area before further processing.

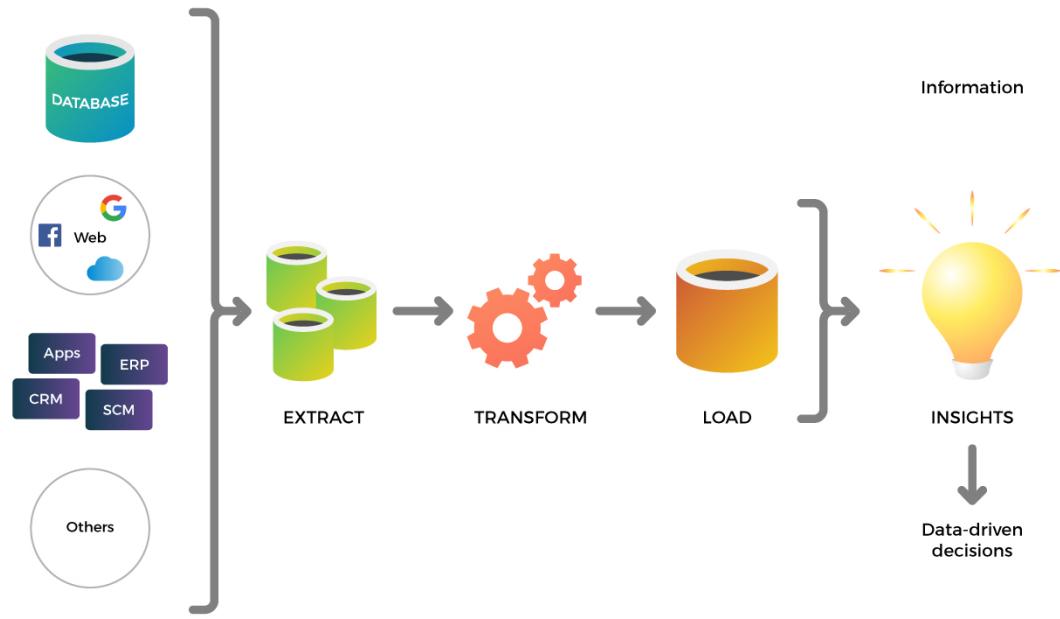
- **Data Transformation (Cleaning Zone):**

An ETL job takes over, accessing the data from the Landing Folder. This job performs essential cleaning tasks on the raw data. This might involve removing duplicates, correcting formatting errors, or filling in missing values. Once cleaned, the processed data is transferred to a new location called the "Cleaned Folder."

- **Data Processing (Processed Zone):**

A separate job titled "Generate Embeddings" focuses on the data within the Cleaned Folder. This job utilizes the cleaned data to create "embeddings." Embeddings are a condensed representation of the data, often used in machine learning applications. The final processed data, including the embeddings, resides in a "Processed Folder." This folder likely serves as the destination for the data pipeline.

6. ETL (Extraction, Transformation, and Load)



6.1 Extraction

The first step of the ETL process is to extract data from the source systems. The primary data source for this project is the official website of the Indian Premier League (IPL), available at <https://www.iplt20.com/>. The IPL website offers in-depth details on all of the teams, players, matches, statistics, and other pertinent information about the IPL matches. Storing extracted raw data in an AWS S3 bucket is a common and scalable approach for data storage, especially when dealing with large volumes of data. Using different tiers like Bronze, Silver, and Gold for structured data processing can help in organizing and prioritizing the data processing tasks based on their importance and complexity.



CLEANED



LANDING



PROCESSED

The Bronze Tier is used for raw and unprocessed data storage. The silver tier is used for data transformation and preliminary analysis. The gold tier is used for advanced analytics, embedding, and reporting.

The data source is a real-time data source, which means that new data is being added to the source dump bucket regularly.

Amazon S3 > Buckets > myybuckett

myybuckett [Info](#)

[Objects](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

Objects (3) [Info](#)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in this bucket.

Find objects by prefix Show versions

<input type="checkbox"/>	Name	Type
<input type="checkbox"/>	<input checked="" type="checkbox"/> cleaned/	Folder
<input type="checkbox"/>	<input checked="" type="checkbox"/> landing/	Folder
<input type="checkbox"/>	<input checked="" type="checkbox"/> processed/	Folder

6.2 Transformation

Transforming the data that has been retrieved from the source systems into a format that is consistent with the target system is the second stage in the ETL process.

Transformations of the text data include the following:

- Removing hyperlinks
- Removing the Twitter mentions
- Removing the emojis and special characters
- Removing the images and image links

AWS Glue: AWS Glue is a serverless data integration service that makes it easy for analytics users to discover, prepare, move, and integrate data from multiple sources. You can use it for analytics, machine learning, and application development. It also includes additional productivity and data ops tooling for authoring, running jobs, and implementing business workflows. AWS Glue consolidates major data integration capabilities into a single service. These include data discovery, modern ETL, cleansing, transforming, and centralized cataloging. It's also serverless, which means there's no infrastructure to manage. With flexible support for all workloads like ETL, ELT, and streaming in one service, AWS Glue supports users across various workloads and types of users.

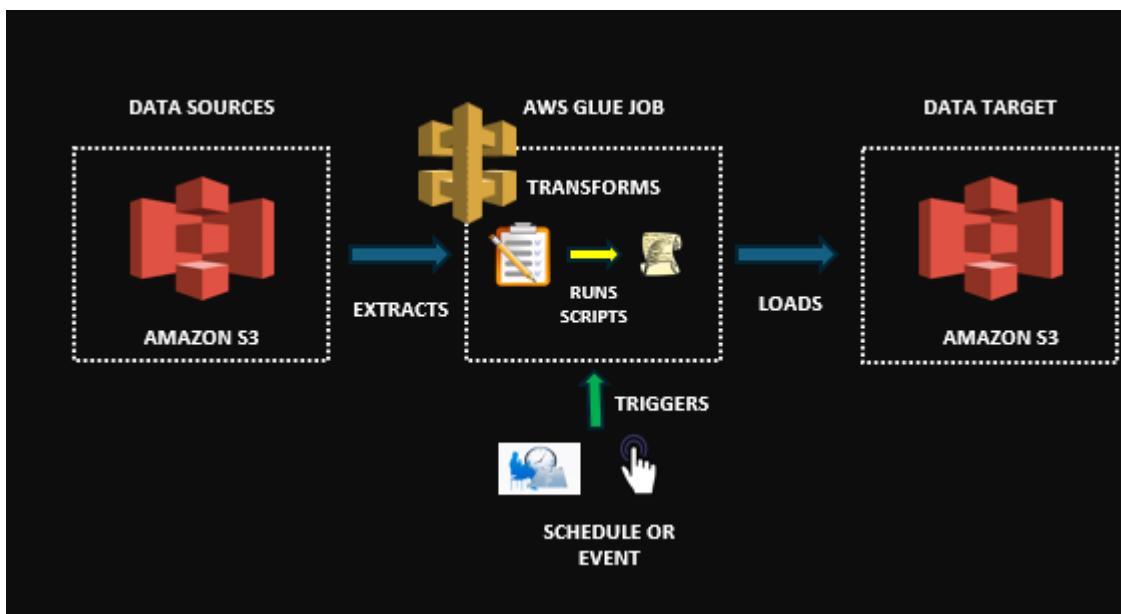
Also, AWS Glue makes it easy to integrate data across your architecture. It integrates with AWS analytics services and Amazon S3 data lakes. AWS Glue has integration interfaces and job-authoring tools that are easy to use for all users, from developers to business users, with tailored solutions for varied technical skill sets. AWS Glue Data Quality allows you to measure and monitor the quality of your data so that you can make good business decisions. AWS Glue Data Quality for AWS Glue ETL jobs lets you perform proactive data quality tasks. Proactive tasks help you identify and filter out bad data before you load a data set into your data lake.

In AWS Glue, you can create Data Catalog objects called triggers, which you can use to either manually or automatically start one or more crawlers or extract, transform, and load (ETL) jobs. Using triggers, you can design a chain of dependent jobs and crawlers.

Your jobs (3) Info			
<input type="checkbox"/>	Job name	Type	Last modified
<input type="checkbox"/>	aws-generate-embeddings	Python shell	8/4/2024, 1:11:35 pm
<input type="checkbox"/>	iplscrapingjob	Python shell	30/3/2024, 9:41:17 pm
<input type="checkbox"/>	iplgluejob	Python shell	29/3/2024, 8:11:00 pm

Process Flow of AWS Glue:

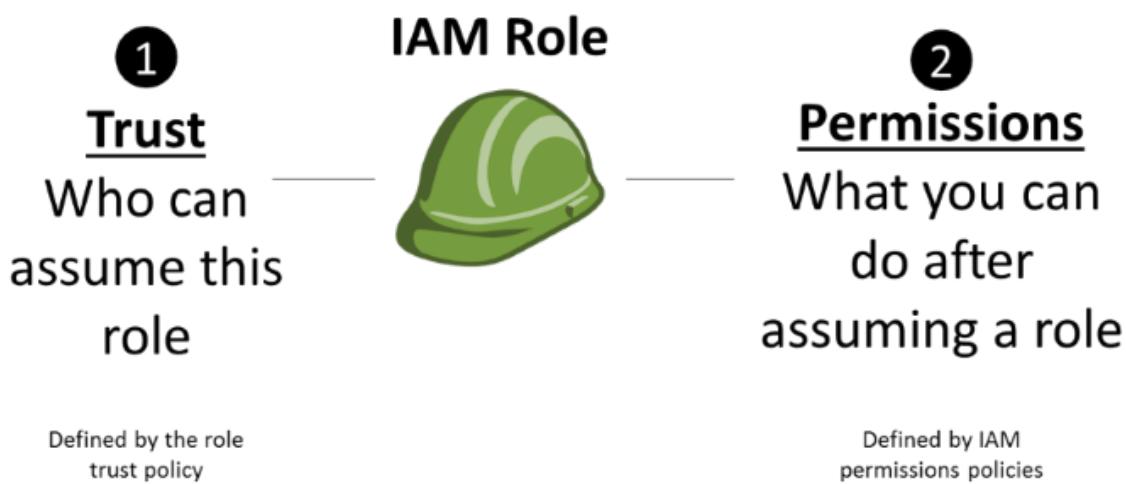
In an AWS Glue job designed for data transformation, Python scripts are utilized to clean and process the data. Additionally, the job is configured to respond to new data uploads on the official website using delta logic and triggers. When new data is detected, these triggers activate the script, initiating the transformation process. Once the data is cleaned and processed, it is then fetched and stored back into an AWS S3 bucket, ensuring that the most recent and accurate data is always available for analysis and other applications.



By following the above steps in data preprocessing, we achieve the following:

- **Data Conformity:** The transformation stage plays a crucial role in ensuring data conformity by standardizing the data format to align with the target system. By removing hyperlinks, Twitter mentions, emojis, special characters, images, and image links, the data is streamlined to meet specific standards. This standardization enhances data consistency across different sources and ensures that the data conforms to predefined rules and formats, thereby facilitating seamless integration and analysis.
- **Data Quality:** AWS Glue's Data Quality capabilities further enhance the quality of transformed data by allowing proactive data quality tasks and monitoring. The cleansing and transformation steps, such as removing unwanted characters and links, contribute to improving data accuracy and reliability. By identifying and filtering out bad data before loading it into the data lake, AWS Glue ensures that only high-quality, trustworthy data is available for analysis and decision-making, thereby enhancing overall data quality.
- **Data Integrity:** Data integrity is maintained through the careful handling and processing of data during transformation. The use of Python scripts and delta logic ensures that the data is processed accurately without loss or corruption. Triggers configured to respond to new data uploads help maintain data integrity by initiating the transformation process only when new data is detected. This approach ensures that the most recent and accurate data is consistently available, preserving data integrity throughout the transformation and storage process.
- **Data consistency:** The transformation process, including data cleansing and standardization, contributes significantly to maintaining data consistency. By removing inconsistencies such as duplicates, unwanted characters, and irrelevant links, the data is made more uniform and consistent across different datasets and sources.
- **Data Unbiasness:** Data sourced from official sources through the transformation process ensures unbiasedness. Using AWS Glue to integrate and transform data from credible sources enhances data reliability and transparency, reducing potential biases and ensuring trustworthy analytics and decision-making.

- **Data Governance:** Data Governance refers to the overall management of the availability, usability, integrity, and security of data used within an organization. IAM roles in AWS are pivotal for finely controlling access to resources, following the principle of least privilege. They enable secure cross-account access, federation, and temporary credentials issuance, fostering collaboration while bolstering security. By assigning roles with granular permissions, organizations ensure that users and services have precisely the access they need, enhancing overall security and compliance within the AWS environment. In the context of IPLGPT chatbot, the IAM (Identity and Access Management) roles are crucial for defining access permissions and controlling interactions within the system. By assigning specific roles to users or entities, administrators can regulate who can perform certain actions, ensuring security and adherence to organizational policies. IAM roles also facilitate the segregation of duties and enable the implementation of the principle of least privilege, enhancing overall security posture within the IPLGPT chatbot ecosystem.



6.3 Load

In the ETL (Extract, Transform, Load) process, the "Load" phase is crucial for ingesting the cleaned and transformed data into a target data repository or data warehouse. During this phase, the processed data from the transformation stage is loaded into the designated storage system, which could be a relational database, a data lake, or cloud storage like AWS S3. The loading process is often optimized to ensure data integrity, consistency, and

performance, utilizing batch or real-time loading mechanisms based on the specific requirements of the data analytics pipeline. The transformed data is stored in the data pipeline in the processed folder after the transformation is performed on it. Proper data loading is essential to make the transformed data readily available for querying, analysis, and reporting, enabling organizations to derive valuable insights and make informed decisions based on up-to-date and accurate data.

- **Data Warehouse:** A centralized repository that stores structured, historical data from multiple sources for reporting and analysis purposes, facilitating decision-making processes within an organization.
- **Data Mart:** A subset of a data warehouse that focuses on a specific department, function, or business unit, containing pre-summarized and formatted data tailored to the needs of a particular group of users.
- **Data Lake:** A storage repository that holds a vast amount of raw data in its native format until it's needed. It allows for flexible and scalable storage of structured, semi-structured, and unstructured data, enabling diverse analytics and processing capabilities.

Data Warehouse Vs Database Vs Data Mart Vs Data Lake



7. Exploratory Data Analysis

Exploratory Data Analysis (EDA) and data visualization are crucial steps in the data analysis process. They help you understand the structure, patterns, anomalies, and relationships within your data, enabling you to make informed decisions and derive meaningful insights.

Data visualization is the graphical representation of data. It helps in conveying complex information in an easy-to-understand manner.

EDA plays a crucial role in the data transformation process as it provides insights into the quality, structure, and characteristics of the data before and after transformation. By conducting EDA, analysts can identify patterns, anomalies, and potential issues in the data, helping to guide the transformation steps effectively. It helps in understanding the distribution of data, detecting outliers, and assessing data quality, which is essential for ensuring data conformity, integrity, and consistency.

7.1 Tableau

Tableau is a powerful data visualization and business intelligence tool that allows you to create interactive and visually appealing dashboards and reports. It's widely used in various industries for data analysis, exploration, and presentation.

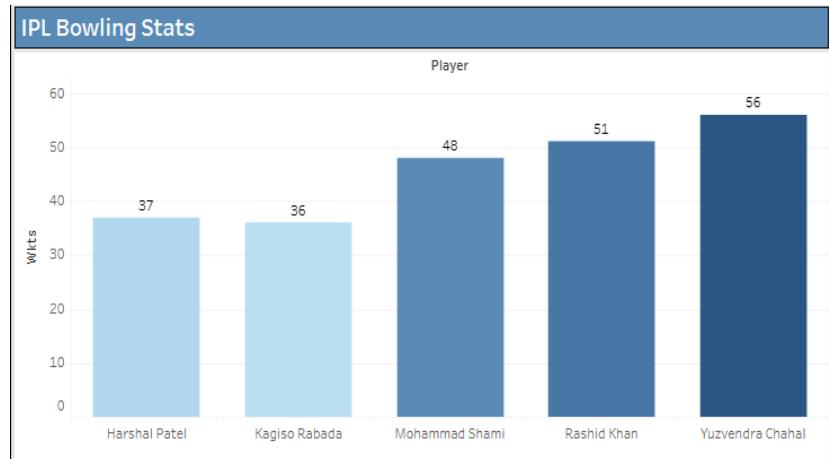
In this tableau dashboard, select ‘All’ years or any particular year (2022/2023/2024) to see the IPL stats for that particular year. Based on the selected year, the Orange cap (player who scored maximum runs) and purple cap (player who took maximum wickets) players are displayed on the dashboard. Additionally, the tournament’s total 6’s and 4’s are calculated and displayed.

The dashboard features a header with team logos: Delhi Capitals, Mumbai Indians, and Kings XI Punjab. Below is a navigation bar with tabs: ORANGE CAP, PURPLE CAP, Tournament 6's, Tournament 4's, and Select Season (with a dropdown for Year). The ORANGE CAP section shows Shubman Gill with 1,556 runs. The PURPLE CAP section shows Yuzvendra Chahal with 56 wickets. The Tournament 6's section shows 2,560. The Tournament 4's section shows 4,789. The Select Season dropdown is set to (All). The bottom row displays player portraits and icons related to the stats.

ORANGE CAP	PURPLE CAP	Tournament 6's	Tournament 4's	Select Season
Shubman Gill 1,556 Runs	Yuzvendra Chahal 56 Wickets	2,560	4,789	Year (All)

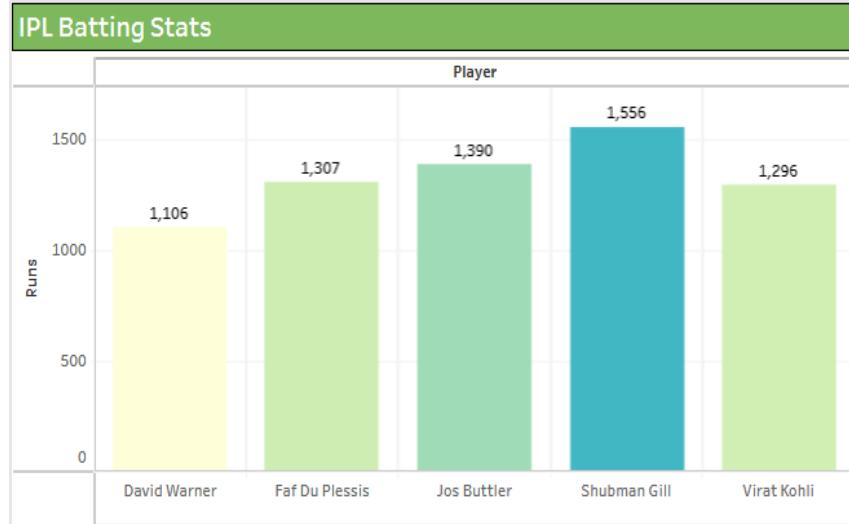
EDA on the bowling data include:

1. The bowler's name.
2. The number of wickets taken by each bowler
3. Top 5 best bowlers



EDA on the batting data include:

1. The batter's name.
2. The number of runs scored by each bowler.
3. Top 5 best batting players



Exploratory Data Analysis (EDA) and data visualization are essential steps in understanding and interpreting complex datasets, enabling insights that drive informed decision-making. In the context of an IPL chatbot, EDA and visualization helped in analyzing player performance, team strategies, and match outcomes, providing valuable data-driven insights to enhance user engagement and interaction.

8. LLM

Large Language Models (LLMs) form a specific category within the broader field of Natural Language Processing (NLP). These models specialize in generating text by analyzing and processing vast datasets. Their notable strength lies in their capacity to comprehend and generate language in a broad and versatile manner. LLMs use something called the transformer model. The transformer model is a neural network that learns context and semantic meaning in sequential data like text. A well-known example of a chatbot using LLM technology is ChatGPT, which incorporates the GPT-3.5 and GPT-4 models.

Problems with Generic LLM

When it comes to Large Language Models (LLMs), there are two possible scenarios involving topics that they may be less knowledgeable about. Firstly, the model may straightforwardly admit that it lacks information on a particular subject because it hasn't been trained on that specific data.

Secondly, there's the potential for what's known as "hallucination", where the model generates responses that are inaccurate or misleading due to its lack of specialized knowledge. This is because generic LLMs are not trained with detailed information in certain areas, such as specific legal rules or medical data, which typically fall outside the scope of a general-purpose LLM's training data.

To address this issue, one method is to fine-tune the model by adding specific data to it and tailor it for needs. However, this report will focus on a simpler approach called RAG, or Retrieval-Augmented Generation.

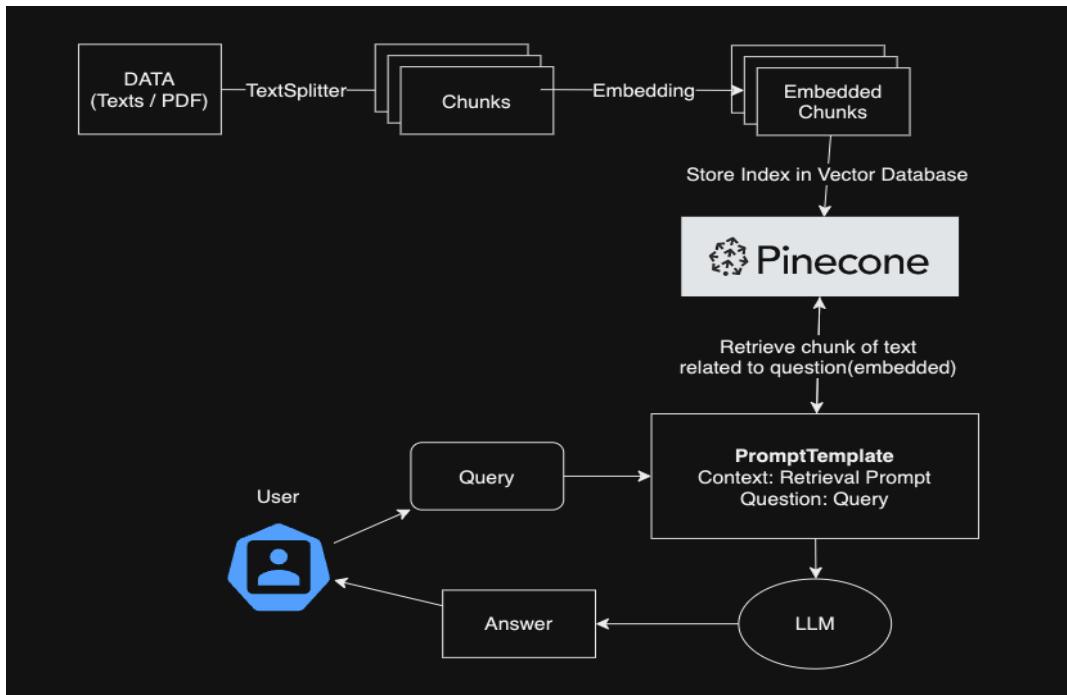
9. RAG Architecture

9.1 Introducing RAG

RAG, short for Retrieval-Augmented Generation, is a way to boost what Large Language Models (LLMs) know by adding more data to them. It's made up of two main components: Indexing: This is about taking in data from various sources and organizing it in a way that the system can easily use, which is indexing. Here we are using text documents scraped from the website iplt20.com which has Match Reports with Date, Teams in the header. Ex - 15 Apr 2024 TATA IPL 2024 Match 30 RCB Vs SRH - Match Report

Retrieval and Generation: To delve deeper into how RAG functions, let's understand its two primary processes: retrieval and generation. The retrieval component acts like a focused search engine, scanning a database (pinecone) of indexed information to find relevant data related to the user's query. This data is then fed into the Large Language Model(GPT-3.5). The model uses this context, along with its trained knowledge base, to generate a response that's more informed and accurate. This synergistic process allows RAG to provide more precise answers by supplementing its extensive but generalized training with specific, targeted information.

In simpler terms, RAG helps LLMs to be more knowledgeable by pulling in extra information when needed to answer questions better. This is how the architecture of the chatbot will look:



9.2 Tools Used:



LangChain is an open-source framework written in Python and JavaScript, designed for building applications centered around language models. LangChain provides components that allow non-AI experts to be able to implement existing AI language models into their applications. This framework is versatile and supports various functions such as text summarization, tagging, and RAG creation.

LLM Model – GPT 3.5 Turbo

GPT-3.5 is a significant upgrade from its predecessor, GPT-3, developed by OpenAI. It boasts a larger model size with more parameters, making it capable of capturing more

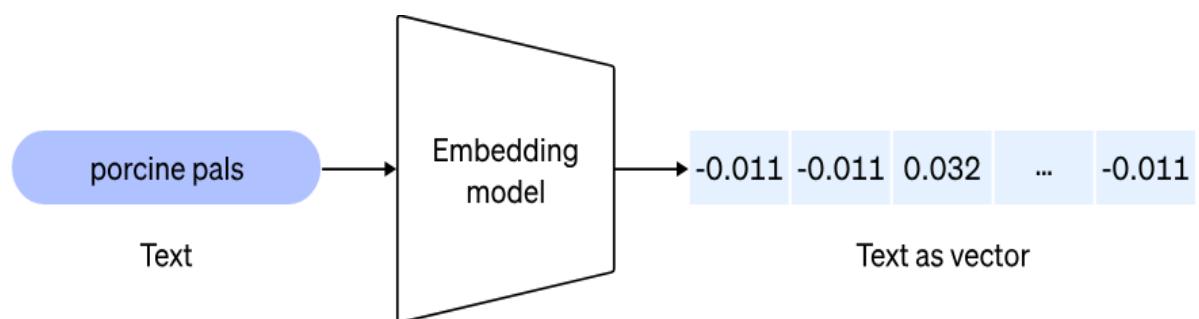
intricate patterns in text data. Built on the Transformer architecture, specifically the GPT-3 architecture, it leverages attention mechanisms and transformer layers for efficient language processing.

One of its key strengths lies in its diverse capabilities. GPT-3.5 can generate human-like text, answer questions, summarize content, translate languages, and perform various natural language processing tasks. Its flexibility extends to fine-tuning, allowing developers to customize its behavior for specific tasks or domains, enhancing its performance in targeted applications.

Furthermore, GPT-3.5 supports continual learning, meaning it can be further trained on new data to adapt and improve its understanding and generation of text over time. OpenAI has also prioritized ethical considerations, implementing measures to mitigate biases and promote responsible AI usage in GPT-3.5's development and deployment.

Embeddings

An embedding is a sequence of numbers that represents the concepts within content such as natural language or code. Embeddings make it easy for machine learning models and other algorithms to understand the relationships between content and to perform tasks like clustering or retrieval. They power applications like knowledge retrieval in both ChatGPT and the Assistants API, and many retrieval augmented generation (RAG) developer tools.



We have used text-embedding-ada-002 model from Open AI which is the most capable 2nd generation embedding model, replacing 16 first generation models. Each Match Report of 6000 characters is represented in 1536 dimensions.

Pinecone: Vector database

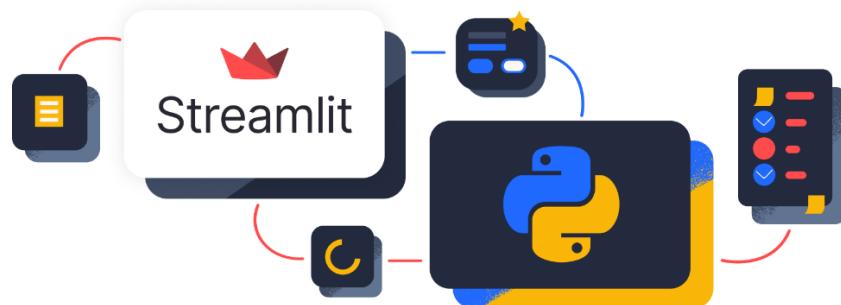


Before we delve into Pinecone, let's clarify what a vector database is. A vector database, such as Pinecone, stores data in the form of vectors, which are arrays of numbers — e.g. [0.1, 3.21, -1.3, 9.2, ...]. This approach allows for efficient similarity searches, as it groups similar data and enables models or applications to retrieve relevant information effectively.

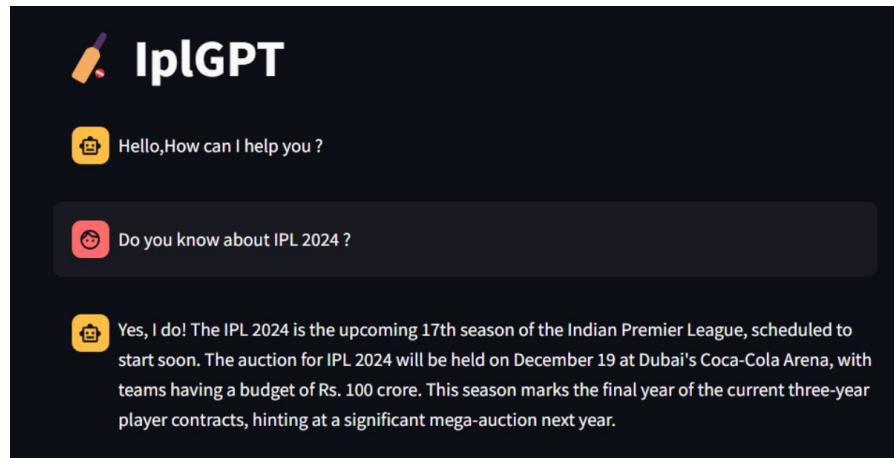
Pinecone is a cloud-based vector database optimized for machine learning applications. It is designed to efficiently store and retrieve dense vector embeddings, making it ideal for enhancing Large Language Models (LLMs) with long-term memory and improving their performance in tasks such as natural language processing. It offers quick data retrieval, ideal for chatbots, and includes a free tier for storing up to 100,000 vectors. Although there are open-source vector databases available like Chroma, Weaviate, and Milvus, Pinecone is preferred for its simplicity and ease of use.

10. Streamlit

Data scientists ought to recollect the usage of Streamlit apps for a multitude of motives. Firstly, Streamlit apps gives a splendid benefit via allowing customers to increase interactive web apps without delving into the complexities of conventional internet improvement. With only a few lines of code, utilizing the `import streamlit`'s command, statistics scientists can effortlessly create captivating internet apps. The “Sweet Streamlit Hello App” serves as a great introduction, showcasing the platform’s intuitive interface and functionalities.



For the ones inside the field of information science, Streamlit provides an invaluable tool for deploying fashions with minimum programming knowledge. Its user-pleasant nature empowers data scientists to proportion their insights and analyses comfortably. Furthermore, within the realm of software creation, offering a seamless user revel in is paramount. Streamlit apps streamline this manner by way of facilitating the rapid improvement of effective user interfaces for information-intensive mobile apps and internet apps alike. By leveraging Streamlit’s abilities, statistics scientists can consciousness on their center understanding in records analysis and model advent, while seamlessly translating their work into person-friendly, interactive programs for sharing with colleagues and stakeholders.



- **Streamlit Interface:** Begin by means of developing a Streamlit net utility. Streamlit affords an intuitive way to build interactive net interfaces with the use of Python. You'll layout the user interface in which customers can input textual content or pick out options.
- **User Input:** Within your Streamlit app, provide enter fields or widgets in which users can enter text or specify their requirements. This will be a text field for getting into activities or questions.
- **Integration Point:** Create an integration point to your Streamlit app where consumer enter is sent to OpenAI's API. This is typically completed through HTTP requests, where you pass the consumers enter to the API endpoint.
- **OpenAI API:** OpenAI offers APIs that permit developers to interact with their language models. Once your Streamlit app sends a request to the OpenAI API, the API methods the input text the usage of the language model and generates a response.
- **Response Handling:** After receiving a reaction from the OpenAI API, your Streamlit app processes the output. This would possibly involve parsing the generated textual content, formatting it for display, or any additional processing required appearing.
- **Display Output:** Finally, your Streamlit app offers the output to the consumer. This can be displayed as text at the web page or incorporated into interactive factors like charts or graphs, relying on the nature of your application.
- **User Interaction:** Users can then interact with the output, provide additional

entrance if necessary, and continue the communique or venture glide inside the Streamlit app.

By integrating OpenAI's language models with Streamlit, you may create dynamic and interactive applications that leverage the strength of natural language processing. This permits an extensive range of applications, from chatbots and digital assistants to textual content summarization gear and creative writing assistants, all available via a person-friendly net interface.

Results:

The screenshot shows a dark-themed AI interface. At the top, there is a red speech bubble icon followed by the text "Tell me about Final match of IPL 2023 , also highlight jadeja's contribution". Below this, a yellow document icon is followed by the text: "In the final match of IPL 2023, Chennai Super Kings (CSK) faced defending champions Gujarat Titans (GT). CSK emerged victorious, winning their fifth IPL title by defeating GT by five wickets via the DLS Method in a thrilling match at the Narendra Modi Stadium in Ahmedabad." Further down, another yellow document icon is followed by: "Ravindra Jadeja played a crucial role in the grandstand finish of the final match. With 10 runs needed off the last two balls, Jadeja provided a spectacular finish by hitting a six and a four to lead CSK to victory. He scored an unbeaten 15 off just 6 balls, showcasing his prowess with the bat and nerves of steel in pressure situations." At the bottom, another yellow document icon is followed by: "Jadeja's contribution, along with the performances of other CSK players like Devon Conway, Ruturaj Gaikwad, Shivam Dube, Ajinkya Rahane, and Ambati Rayudu, helped CSK successfully chase down the revised target of 171 runs in 15 overs after rain interrupted the match." Finally, at the very bottom, another yellow document icon is followed by: "Overall, Jadeja's heroic performance in the final match was instrumental in CSK's victory and their fifth IPL title win."

Following output can be described and understood by analyzing different components:

Retrieval:

IPLGPT retrieve relevant information based on keywords like "IPL," "Chennai Super Kings," "Gujarat Titans," "Ravindra Jadeja," and "final match."

It has access to pinecone vector database, and it fetches the relevant document related to the user input by performing cosine similarity search.

Generation:

Based on the retrieved information, the IPLGPT attempts to generate a summary of the event by providing context from the above step to the LLM model which in this case is GPT-3.5 Turbo.

It would identify key points like the winner, loser, margin of victory, and the player who

played a crucial role that it fetches from the documents.

Overall Perspective:

The RAG chatbot would understand that the output describes a significant sporting event - the IPL final.

It would recognize CSK's victory and highlight Jadeja's contribution to their win.

Depending on the document vector data, it might offer additional details or analysis.

11. AI Agents

11.1 What are AI Agents?

An artificial intelligence (AI) agent is a software program that can interact with its environment, collect data, and use the data to perform self-determined tasks to meet predetermined goals. Humans set goals, but an AI agent independently chooses the best actions it needs to perform to achieve those goals. For example, consider a contact center AI agent that wants to resolve customer queries. The agent will automatically ask the customer different questions, look up information in internal documents, and respond with a solution. Based on the customer responses, it determines if it can resolve the query itself or pass it on to a human.

What are the key principles that define AI agents?

All software autonomously completes different tasks as determined by the software developer. So, what makes AI or intelligent agents special?

AI agents are rational agents. They make rational decisions based on their perceptions and data to produce optimal performance and results. An AI agent senses its environment with physical or software interfaces.

For example, a robotic agent collects sensor data, and a chatbot uses customer queries as input. Then, the AI agent applies the data to make an informed decision. It analyzes the collected data to predict the best outcomes that support predetermined goals. The agent also uses the results to formulate the next action that it should take. For example, self-driving cars navigate around obstacles on the road based on data from multiple sensors.

How does an AI agent work?

AI agents work by simplifying and automating complex tasks. Most autonomous agents follow a specific workflow when performing assigned tasks.

Determine Goals

The AI agent receives a specific instruction or goal from the user. It uses the goal to plan tasks that make the outcome relevant and useful to the user. Then, the agent breaks down

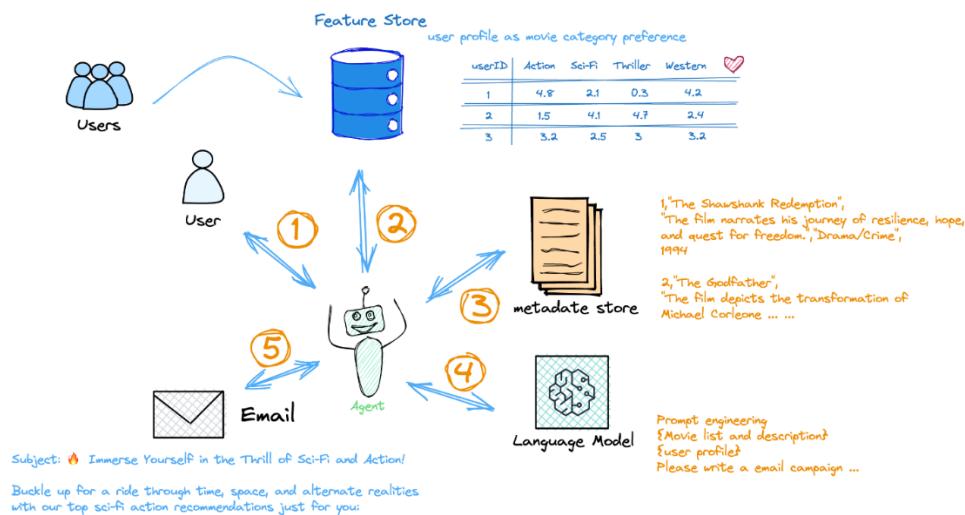
the goal into several smaller actionable tasks. To achieve the goal, the agent performs those tasks based on specific orders or conditions.

Acquire Information

AI agents need information to act on tasks they have planned successfully. For example, the agent must extract conversation logs to analyze customer sentiments. As such, AI agents might access the internet to search for and retrieve the information they need. In some applications, an intelligent agent can interact with other agents or machine learning models to access or exchange information.

Implement Tasks

With sufficient data, the AI agent methodically implements the task at hand. Once it accomplishes a task, the agent removes it from the list and proceeds to the next one. In between task completions, the agent evaluates if it has achieved the designated goal by seeking external feedback and inspecting its own logs. During this process, the agent might create and act on more tasks to reach the outcome.



11.2 Our AI Agent - IPLGuru

IPLGuru is an advanced virtual assistant designed to streamline tasks and enhance data analysis capabilities. It integrates cutting-edge technologies to offer two main tools that significantly contribute to productivity and decision-making processes.

Tool 1: Interacting with the System and Note-Taking:

IPLGuru's first tool is a robust system interaction module coupled with intelligent note-taking functionalities. This tool allows seamless interaction with systems such as personal computers, facilitating data capture, organization, and communication. Additionally, IPLGuru's note-taking feature is designed to capture important information, create structured notes, and facilitate collaboration. This functionality acts as a demonstration of agents' ability to integrate diverse external systems and facilitate interaction.

Tool 2: Answering Data Questions with PandasQueryEngine:

IPLGuru's second tool leverages the powerful PandasQueryEngine to answer complex questions about structured data files. PandasQueryEngine is a data manipulation and analysis tool that IPL Guru utilizes to provide accurate and insightful responses to data-related queries. By harnessing PandasQueryEngine's capabilities, AI Agent IPLGuru can perform data filtering, sorting, and aggregation tasks seamlessly. It can handle large datasets efficiently, extract relevant information, and present meaningful insights to users.

Integration and Synergy:

These two tools within IPLGuru work synergistically to enhance overall functionality and user experience. For example, users can interact with the system to retrieve data, and IPLGuru can then analyze this data using PandasQueryEngine to provide relevant answers to queries.

The integration of these tools streamlines workflows, reduces manual data processing efforts, and enables users to make data-driven decisions more effectively. IPLGuru's natural language interaction capabilities and seamless integration with existing systems make it an asset for businesses and professionals.

Real-World Applications:

IPLGuru's core technology can be deployed in various real-world scenarios, including data analysis projects, project management tasks, and research initiatives with just change in data source. In analytics project, IPLGuru's technology can be helpful to analyze customer data, identify trends, and optimize marketing strategies based on insights generated.

Results –

Tool 1 – Note Engine

```
(ai-agents) shravan@LAPTOP-JR3B3TTQ:~/RAG/RAG+AI-Agents/PythonAgentAI-main$ python3 main.py
Enter a prompt (q to quit): Make a note saying this is agent interacting with system
Thought: I can use the note_saver tool to make a note.
Action: note_saver
Action Input: {'note': 'This is the agent interacting with the system.'}
Observation: note saved
Thought: I can answer without using any more tools.
Answer: Note has been saved successfully.
Note has been saved successfully.
```

Here's a breakdown of the output:

- The script starts by entering a prompt (Enter a prompt (q to quit) :)
- The user then enters a prompt:" Make a note saying this is agent interacting with system."
- The AI agent then thinks about how to complete this task (Thought: I can use the note_saver tool to make a note.)
- The AI agent then acts and uses the note_saver tool (Action: note_saver)
- The AI agent provides additional information about the action ('Action Input: {'note': 'This is the agent interacting with the system. '}~)
- The AI agent then observes that the note has been saved successfully (Observation: note saved)
- Finally, the AI agent confirms that the note has been saved successfully (Thought: I can answer without using any more tools. Answer: Note has been saved successfully.)
- Overall, the screenshot appears to show a successful interaction between a user and an

AI agent where the agent can use a tool to save a note as instructed.

Tool 2 – PandasQueryEngine

```
Enter a prompt (q to quit): Can you tell me who are the top 3 highest scoring batsmen in IPL 2023 ?
Thought: I can use the runs_data tool to get the information about the top 3 highest scoring batsmen in IPL 2023.
Action: runs_data
Action Input: {'input': 'top_scorers'}
> Pandas Instructions:
...
df.sort_values(by='Runs', ascending=False).head()
> Pandas Output:
          Player      Team Name  Mat  Inns  NO  Runs   HS   Avg   BF   SR  100  50   4s   6s  Year
110  Shubman Gill  Gujarat Titans  17   17   2   890  129  59.33  564  157.80  3   4   85  33  2023
276  Jos Buttler  Rajasthan Royals  17   17   2   863  116  57.53  579  149.05  4   4   83  45  2022
111  Faf Du Plessis  Royal Challengers Bangalore  14   14   1   730  84  56.15  475  153.68  0   8   60  36  2023
112  Devon Conway  Chennai Super Kings  16   15   2   672  92*  51.69  481  139.70  0   6   77  18  2023
113  Virat Kohli  Royal Challengers Bangalore  14   14   2   639  101*  53.25  457  139.82  2   6   65  16  2023
Observation:
          Player      Team Name  Mat  Inns  NO  Runs   HS   Avg   BF   SR  100  50   4s   6s  Year
110  Shubman Gill  Gujarat Titans  17   17   2   890  129  59.33  564  157.80  3   4   85  33  2023
276  Jos Buttler  Rajasthan Royals  17   17   2   863  116  57.53  579  149.05  4   4   83  45  2022
111  Faf Du Plessis  Royal Challengers Bangalore  14   14   1   730  84  56.15  475  153.68  0   8   60  36  2023
112  Devon Conway  Chennai Super Kings  16   15   2   672  92*  51.69  481  139.70  0   6   77  18  2023
113  Virat Kohli  Royal Challengers Bangalore  14   14   2   639  101*  53.25  457  139.82  2   6   65  16  2023
Thought: (Implicit) I can answer without any more tools!
Answer: Observation: The top 3 highest scoring batsmen in IPL 2023 are:
1. Shubman Gill from Gujarat Titans with 890 runs.
2. Jos Buttler from Rajasthan Royals with 863 runs.
3. Faf Du Plessis from Royal Challengers Bangalore with 730 runs.
Observation: The top 3 highest scoring batsmen in IPL 2023 are:
1. Shubman Gill from Gujarat Titans with 890 runs.
2. Jos Buttler from Rajasthan Royals with 863 runs.
3. Faf Du Plessis from Royal Challengers Bangalore with 730 runs.
Enter a prompt (q to quit): []
```

Here's a breakdown of the output:

- **User prompt:** The script begins by showing the user prompt: "Enter a prompt (q to quit): Can you tell me who are the top 3 highest scoring batsmen in IPL 2023?"
- **Agent thought process:** The AI agent then thinks about how to answer the question by stating: "I can use the runs data tool to get the information about the top 3 highest scoring batsmen in IPL 2023."
- **Agent action:** The AI agent then acts and uses the "runs_data" tool, referring to a code module designed to access and process Ipl data.
- **Action input:** The input for this action is a dictionary containing a key named "input" with the value "top_scorers" specifying that the agent is looking for the top scorer's data.
- **Pandas dataframe output:** The response from the "runs_data" tool is likely a Pandas dataframe containing information about the batsmen. The dataframe output includes columns like "Player", "Team", "Mat", "Runs", "HS", "Avg", "BF", "SR", "100", "50", "4s", and "6s". It appears the data is partially shown here, truncated at "Year".

- **Agent observation:** The agent then observes the data and identifies the top 3 batsmen based on their "Runs" scored. It seems there might be inconsistencies in the data as it observes "Shubman Gill" at number 110 with 898 runs while mentioning him again at number 1 with 890 runs. Additionally, it observes "Faf Du Plessis" at number 111 with 730 runs and again at number 3.
- **Agent Answer:** Ignoring the inconsistencies, the AI agent concludes by stating the top 3 scorers are "Shubman Gill" from Gujarat Titans with 890 runs, "Jos Buttler" from Rajasthan Royals with 863 runs, and "Faf Du Plessis" from Royal Challengers Bangalore with 730 runs.

12. My Conclusion

Collaborating on the IPL GPT project with my esteemed team was an exhilarating journey that expanded the horizons of my big data analytics expertise. Diving into the world of sports analytics, our team, “CHAIGPT” set out to redefine user engagement dynamics within the IPL community.

Through my involvement in the IPL GPT initiative, I gained expertise in data extraction using Beautiful Soup. This Python library proved invaluable for efficiently parsing HTML and XML files. I learned to navigate HTML structures, identify relevant elements, and extract desired data using methods like `find()` and `find_all()`. Additionally, I mastered techniques for handling dynamic content and ensuring robustness in data extraction processes. These skills enabled me to retrieve structured data from web pages effectively, optimizing performance and adapting to changes in website layouts. Overall, my experience with Beautiful Soup has equipped me with essential tools for harnessing web data in various projects.

Then, analyzing this AWS Glue architecture was an eye-opener. It solidified my understanding of ETL, highlighting the clear separation between extracting, transforming, and loading data. AWS Glue seems like a game-changer for orchestrating these workflows on AWS. The architecture also emphasizes the power of Amazon S3 as a data lake, storing both raw and processed data. Finally, it underscores the importance of data transformation before loading, making the data truly analysis ready.

Through my exploration of the Representational AI (RAG) architecture, I've gained invaluable insights into the intricacies of building intelligent systems for text retrieval and summarization. Understanding how data flows through various stages, from retrieval to preprocessing, embedding, storage, retrieval, and summary generation, has deepened my appreciation for the complexity of such systems. I've learned about the importance of efficient data storage and retrieval mechanisms, exemplified by Pinecone's fast similarity search capabilities. Moreover, witnessing the integration of

cutting-edge technologies like OpenAI's text-embedding model and GPT-3.5 for preprocessing and summary generation has broadened my understanding of their practical applications. The role of the Streamlit as the interface for user interaction highlighted the significance of seamless integration and user experience in deploying AI-powered solutions. Overall, this experience has equipped me with valuable knowledge and insights into the architecture and components of AI systems tailored for text retrieval and summarization, underscoring the transformative potential of AI in delivering relevant and concise information to users.

Dealing with vast amounts of data posed significant challenges during the development of the AI model. Initially, I attempted to utilize data spanning from 2008 onwards to train the model comprehensively. However, I encountered issues where the model struggled to provide accurate and relevant responses. It seemed to generate random answers, indicating confusion or lack of coherence. Recognizing the limitations imposed by the sheer volume of data, I made the decision to limit the dataset to the years 2022-2024. This focused approach allowed for better control over the data quality and relevance, leading to more consistent and reliable responses from the model.

Additionally, I faced the challenge of automating daily tasks efficiently, particularly in the context of data processing and analysis. To address this, I implemented automation using AWS Glue, a service designed for data integration, transformation, and scheduling. However, scheduling daily automation tasks posed its own set of challenges, especially with time zone considerations. Given the requirement for automation to run from 11am to 11am EST, ensuring seamless execution across different time zones and handling potential issues related to time discrepancies became crucial aspects to manage. Through careful planning and adjustments, I successfully implemented a solution that facilitated daily automation while effectively addressing time zone complexities.

The project's triumph speaks volumes about the potency of analytics in reshaping how users engage, firmly cementing my conviction in the boundless potential of data-driven solutions to redefine user experiences within online communities. It was a journey

marked by our commitment to inclusivity, fostering a culture of openness, and nurturing a collaborative environment where everyone's strengths were valued and leveraged for maximum impact.

Our endeavors, deeply rooted in innovation and propelled by a fervent dedication to analytics, underscore the infinite possibilities when technology synergizes with human-centered aspirations. It's been a voyage of exploration, creativity, and, above all, a testament to the transformative influence of data analytics. In summary, I've personally honed skills in data management, machine learning, natural language processing, and visualization, harnessing state-of-the-art technologies such as Beautiful Soup, Lang chain, OPEN AI, Pinecone, AWS, and RAG (LLM). I've developed the acumen to tackle intricate data challenges, from ensuring data integrity to predicting user behavior and optimizing model performance. Moreover, I've cultivated a knack for fostering teamwork, effective communication, and the adeptness to harness diverse talents towards shared objectives.

Recognizing the imperative of ethical data usage, bias mitigation, and privacy preservation in analytics, I've embraced a culture of perpetual learning, embracing new tools and methodologies, and adapting swiftly to evolving project dynamics. The key insights gleaned, and the business values generated underscore not only the enrichment of my technical prowess but also the tangible contributions made to the business's bottom line.

The journey has been transformative, not just in terms of technical expertise but also in delivering palpable benefits to the organization, leaving an indelible mark on my professional growth.

13. Reference

1. <https://beautiful-soup-4.readthedocs.io/en/latest/>
2. [What is AWS Glue? - AWS Glue \(amazon.com\)](#)
3. [AWS Step Functions Documentation \(amazon.com\)](#)
4. [AWS SDK for Python \(Boto3\) Documentation \(amazon.com\)](#)
5. <https://python.langchain.com/doc>
6. <https://chat.openai.com/>
7. <https://docs.s.io/>
8. <https://platform.openai.com/docs/api-reference>
9. <https://www.youtube.com/@underfitted>
10. <https://www.youtube.com/@freecodecamp>
11. <https://platform.openai.com/docs/guides/embeddings>