p

10 Academy Cohort B - Weekly Challenge: Week 3

## Redash chatbot add-on: LLM based chatbot for Advanced Data Analytics, Visualisation, and Automated Insight Extraction

# Overview

Business Need

Our company is seeking to significantly enhance its data analysis capabilities, specifically focusing on comprehensive YouTube data exploration. ***The aim of this project is to build a novel Redash chat add-on that our team members can use to extract insight from multiple Redash dashboards and from connected databases using natural language.*** The chat add-on enables a seamless conversation in a question and answer format and autonomous knowledge discovery. User queries could be about what is already displayed in the dashboard or questions that require generating SQL query using LLMs to be run against our connected databases. The end-to-end system you will help us build will empower and allow our team members to extract deep, meaningful, and actionable insights from our business intelligence (BI) platforms.

Our company's BI dashboards are not only to give us monitoring capability of our business process, but also to help us transform the data we collect from multiple systems such as YouTube, Slack and Gmeet into actionable insights that can drive strategic decisions and offer a competitive edge in understanding digital content consumption trends.

The scope of this project extends to developing a Redash add-on in the frontend and an intelligence backend that translates user queries into one of the following

* Summary of visualisations in the current dashboard
* Insight about data returned by existing SQL queries
* Auto generate SQL queries and visualisations
* Auto generate new Redash dashboards from existing and auto generated SQL queries and its associated visualisations.

This tool will be a game-changer to BI and data analysis using Redash - helping translate natural language queries into complex SQL queries. This functionality is expected to democratize data analytics, allowing team members with non-technical backgrounds to easily extract and visualize data without deep knowledge of SQL. The integration of this add-on with Redash will streamline our analytical processes, making data exploration more accessible, efficient, and user-friendly.

Additionally, the backend API you will develop could be usable in many data analysis and BI projects.

# Data

* Video metadata for all videos uploaded in our YouTube channel
* Time Series viewership metadata for all videos uploaded in our YouTube channel
* Time Series comments for all our videos uploaded in our YouTube channel
* Time Series transcribed text for selected videos uploaded in our channel
* This week [data](https://drive.google.com/drive/folders/1kpqrsliBsUchbTs3cEccRXYGNyGaGWeY?usp=drive_link)

# Expected Outcomes

## Skills:

* Data Analysis and Visualization:
  + Proficiency in analysing time series data
  + Proficiency in EDA techniques.
  + Expertise in creating intuitive and informative dashboards.
* Programming and Development Skills:
  + Proficiency in Python and Javascript (React) programming languages
  + Proficiency in SQL
  + Proficiency in understanding code bases for complex software packages
* Proficiency in Prompt Engineering
* Proficiency in using OpenAI API
* Experience in developing add-ons or plugins
* SQL and Database Management
* Experience in running concurrent tasks using Celery
* Experience in deploying complex software package using Docker and docker-compose
* UI/UX Design

## Knowledge:

* Natural Language Processing (NLP) Knowledge:
* Use of vector databases
* Machine Learning and AI Knowledge

# Team

Instructors:

* Yabebal
* Emtinan
* Rehmet

# Key Dates

* **Discussion on the case** - 9:00 UTC time on **07 May 2024**. Use #all-week-3 to ask questions.
* **Interim Submission** - 8:00 PM UTC time on **Wednesday 08 May 2024**.
* **Final Submission** - 8:00 PM UTC time on **Saturday 11 May 2024**

# Leaderboard for the week

There are 100 points available for the week.

20 points - community growth and peer support.

13 points - technical public and group-based RC channels

* Total number of messages (5)
* Total number of Mentions (3)
* Total number of DM connections (5)

7 points - community activities

* Number of messages in non-technical channels (4)
* On-time presence in Gmeet sessions (3)

30 points - presentation and reporting.

15 points - interim submission. PDF

15 points for the final submission. Blog entry or PDF with 5-8 pages.

50 points - Technical content

20 points - Interim submission

1. Github link submission (20)

30 points - Final submission

* Github Link submission (25)

Badges

Each week, one user will be awarded one of the badges below for the best performance in the category below.

In addition to being the badge holder for that badge, each badge winner will get +20 points to the overall score.

**Visualization** - the quality of visualizations, understandability, skimmability, choice of visualization

**Quality of code** - reliability, maintainability, efficiency, commenting - in the future this will be [CICD](https://en.wikipedia.org/wiki/CI/CD)

**An innovative approach to analysis** -using latest algorithms, adding in research paper content and other innovative approaches

**Writing and presentation** - clarity of written outputs, clarity of slides, overall production value

**Most supportive in the community** - helping others, adding links, tutoring those struggling

The goal of this approach is to support and reward expertise in different parts of the Machine learning engineering toolbox.

# Group Work Policy

This week, you are expected to complete the project with your assigned group. In the table below, your name is assigned to one of the groups we formed.

| Group Name (number) | Group Members |
| --- | --- |
| Group 1 | Henock Dessalegn  Abdelrhman Yasir  Samson Leul |
| Group 2 | Joseph Njuguna  Abraham Sahile  Grace Ogundaini |
| Group 3 | Dawit Wubale  Blen Sleshi  Abraham Teka |
| Group 4 | Daisy Cherono  Getachew Abebe Agegnehu  Mikias Wondim Gebre Medhin |
| Group 5 | Addisu Alemu  Abubeker Shamil  Selam Yoseph |
| Group 6 | Biniyam Teshome  Koomi Toussaint Amoussouvi  Eyerusalem Admassu |
| Group 7 | Sheila Murugi  Temesgen Gebreabzgi  Michael George |
| Group 8 | Gilbert Tcheugoue Djissitchedjibril  Grace Nyutu  Mistir Nigusse |
| Group 9 | Yohanes Teshome Kebede  Mahbubah Abdulhakim  Fatai Abdusalam |
| Group 10 | Dawit Mulie  Bethelhem Mebratu  Hillary Kipkemoi |
| Group 11 | Ermias Asmare  Wandera Martin  Melaku Alehegn |
| Group 12 | Dereje Hinsermu Senbatu  Hanna Mengistu  Jabez Kassa |
| Group 13 | Tewodros Cheru  Selamawit Tibebu  Mihret Kochito Wolde |
| Group 14 | Nyamusi Moraa  Beza Mesfin  Jolly Omere |

# Late Submission Policy

Our goal is to prepare successful learners for a global level job. At work, deadlines are sometimes very strict - either you do it before the deadline or the company loses a substantial opportunity. Moreover, the late communication behaviour (submission in 10 Academy can be considered as progress communication to team leads), blinds team leads and CEOs and is very determinantal in hindering the success of the company.

We have set our late submission as follows

* Submissions are accepted only within the 12 hrs window - 17:00 UTC - 7:00 UTC of the submission deadline
* Frequently late submissions (exceeding 6 total late submissions) will disqualify a person from the list of trainees 10 Academy recommends to partner employers.
* Badges will be rewarded for the cumulative on-time appearances (gmeet calls, on-time assignment submissions, and other places where being on-time is important)

Instructions

The fundamental tasks in this week’s challenge are the following

The workflow for this week's challenge is as follows

* Understand Redash and Data Exploration: Gain a comprehensive understanding of Redash's capabilities and data exploration requirements.
* Create a Database Schema: Design a schema that is well-suited for YouTube data and analytics.
* Outline implementation: Develop an implementation plan for the Redash add-on and backend systems.
* Plan Integration of Large Language Models: Design the integration approach for large language models for natural language understanding.
* Plan your work and set up a development environment to assist in completing the project
* Build frontend system
  + Improve starter Redash chat add-on and make it Robust. You could try the following if you have time
    - Chat window is available in Redash query editor
    - Temporary chat window pops up next to any visualisation in the Redash dashboard. Context for this temporary chat window is the data and query that generated the given visualisation.
* Build Backend System Development,
  + Build a system that translates Natural Language questions in English to SQL
  + Allow generation of relevant visualisation from existing or auto generated SQL queries
  + Allow generation of Redash dashboards from natural language query
* Set up a GitHub repo, integrate unit testing and CICD for proper code package test and deployment

## 

## Possible Work Plan

### Stage 1: Basic Solution with Python and OpenAI API

- **Objective**: Establish foundational chatbot capabilities for interpreting and translating natural language queries into SQL.

- **Python and OpenAI API**: Utilize Python's robust data processing abilities and the OpenAI API for initial natural language understanding and SQL translation.

- **Implementation**: Develop Python scripts for parsing YouTube data, generating SQL queries, and interfacing with Redash for visualizations. This stage addresses primary tasks like data exploration, schema creation, and basic language to SQL translation.

### Stage 2: Enhanced NLP with LangChain Integration

- **Objective**: Improve the chatbot's NLP capabilities to process more complex queries and provide accurate SQL translations.

- **LangChain**: An intuitive open-source framework that enhances the functionality of large language models (LLMs), such as OpenAI or Hugging Face, for dynamic, data-responsive applications.

- **Implementation**: Embed LangChain into the Python backend, replacing basic OpenAI API usage. This will enhance natural language understanding and processing, allowing for sophisticated reasoning and context-aware interactions with data.

### Stage 3: Advanced Data Handling with LLamaIndex

- **Objective**: Optimize data management for more efficient data retrieval and processing.

- **LLamaIndex**: A data framework designed for LLM-based applications, offering tools for data ingestion, indexing, and querying, and facilitating the integration of various data sources.

- **Implementation**: Implement LLamaIndex in the backend to streamline data access. It will enhance the chatbot’s efficiency in data retrieval and query processing, particularly for the diverse data types associated with YouTube analytics.

### Stage 4: Incorporation of Vector Databases for Semantic Search

- **Objective**: Enhance the chatbot’s semantic search capabilities for more relevant and contextual data retrieval.

- **Vector Databases**: Specialized databases designed to handle high-dimensional vector data, crucial for semantic searches and understanding data in terms of semantic similarities.

- **Implementation**: Utilize vector databases to store and search transcribed text, comments, and metadata from YouTube. This will significantly improve the chatbot’s ability to deliver insightful responses based on semantic understanding.

## Task 1: Review the LLM Revolution and Plan your Work

There has been so much development in the area of Large Language and Multi-modal (text, image, audio, video) Models. It is paramount to have a basic understanding of key developments in this space to be able complete the current project.

Ensure you understand the following key topics and tools

[OpenAI Tools](https://platform.openai.com/docs/introduction)

* OpenAI Chat Completion using ChatGPT (the first tool that starts the LLM Revolution)
* OpenAI Assistants, Threads, and Run
* OpenAI Function Call
* OpenAI Advanced Data Analysis
* OpenAI Code Interpreter
* OpenAI Actions (previously called Plugins)

[LangChain Components](https://python.langchain.com/docs/integrations/components)

* LangChain Tools
* LangChain Agents   
  LangChain Memory
* LangChain Retrievers
* LangChain Adapters

[LLamaIndex Components](https://docs.llamaindex.ai/en/latest/#)

* LLamaIndex Data connectors
* LLamaIndex Query functions
* LLamaIndex Indexing

[Vector Databases](https://python.langchain.com/docs/integrations/vectorstores)

* Embedding Documents (image, text, audio, video)
* Semantic Search using Vector Similarity

Browse the the following links and summarise your understanding in a report

1. SQL generation using LLMs: [Can LLMs Replace Data Analysts? Getting Answers Using SQL](https://towardsdatascience.com/can-llms-replace-data-analysts-getting-answers-using-sql-8cf7da132259)
2. Openai-cookbook examples e.g [Backtranslation\_of\_SQL\_queries.py](https://github.com/openai/openai-cookbook/blob/main/examples/Backtranslation_of_SQL_queries.py)
3. Experiment quickly using [Flowise AI](https://flowiseai.com/)
4. Paper to read: [InsightPilot: An LLM-Empowered Automated Data Exploration System](https://arxiv.org/pdf/2304.00477v2.pdf)
5. [LLMs and SQL Langchain Blog](https://blog.langchain.dev/llms-and-sql/)
6. [Awesome collection of tools, models, and ideas in the LLM Arena](https://github.com/underlines/awesome-ml/blob/master/llm-tools.md)
7. [Llamaindex: An Imperative For Building Context-Aware Llm-Based Apps](https://www.leewayhertz.com/llamaindex/)

## Task 2: Tool understanding and Data exploration

* Understanding Redash and Data Exploration:
  + Learn about Redash's capabilities for data visualization and analytics.
  + Explore how Redash can connect to various data sources, including YouTube analytics .
* Create a Database Schema for YouTube Data and Analytics:
* Design a schema to efficiently store and query YouTube data.https://docs.google.com/spreadsheets/d/1amkmUHkU06z\_UaDMr9ROm7FNreRpAti\_uzvSPISMJoY/edit?usp=sharing
* Consider aspects like channel performance, user base, and video expenses.

## Task 3: Building Redash Chat Add-on

* Outline the Architecture for the Redash Add-on and Backend Systems:
  + Plan the structure of the [Redash add-on](https://github.com/RYees/redash-chatgpt-plugin) and how it integrates with backend systems.
  + Develop and Implement the Dashboard Interface with the add-on
  + Design the UI of the dashboard, focusing on user-friendliness for Redash add-on.
  + Implement the design using React and integrate it with Redash.
* Develop the Backend System for Data Storage and Processing:
  + Build a system to store and process data, aligning with the designed database schema.
  + Ensure efficient data retrieval and processing capabilities.
  + For your python backend Framework, you may use [Quart](https://quart.palletsprojects.com/en/latest/index.html) - the lightweight async version of Flask

## Task 4: LLM Understanding and Integration

* Integrate Large Language Models for Natural Language Understanding:
* Incorporate language models to interpret and process natural language queries.
* Focus on translating these queries into actionable data insights.
* Integrate the Add-on to Convert Natural Language Queries into SQL
* Ensure accurate translation and compatibility with the Redash environment.

## Task 5: Automatic Dashboard Generation

* Implement automatic visualisation generation based on user queries, generated SQL queries, and existing visualisation context.:
* Implement creating Redash dashboards using a collection of visualisations .

## Task 6: Blog Reporting

Write a blog-like report that details the process followed, challenges faced. dashboard building process, LLM integration and lessons learnt from this week’s challenge.

**N.B for reporting**

Your report should start with the Introduction, the overall body of your report, and then a Conclusion.

# 

# Tutorials Schedule

In the following, the colour **purple** indicates morning sessions, and **blue** indicates afternoon sessions.

## Tuesday: Understanding Redash

Here the trainees will understand the week’s challenge.

* Introduction to Week Challenge (Yabebal)
* Introduction to Building Redash Add-on using React (Rehmet)

Key Performance Indicators:

* Understanding week’s challenge
* Understanding the ‘what is’ Redash
* Ability to reuse previous knowledge

## Wednesday: Redash source code review and setup

Here the trainees will understand the components of chatbot and data analysis using OpenAi, LangChain, and LIDA.

* Prompt Engineering tools and ideas (Emtinan)
* Building chatbot Backend using LangChain (Emtinan)

## Thursday: LLM usage and Integration

Here the trainees will understand how advanced data analysis and automatic sql query generation .works using LLMs

* Asynchronous execution for OpenAI API Streaming and Redash query execution (Rehmet)
* How ADA, LIDA, AutoGPT, InsightPilot, and other LLM agents work (Rehmet)

## Friday: Concurrent execution for Robust Chatbot

Here the trainees will understand how to do aysnc request to OpenAI API and execute SQL queries in Redash Celerey executer

* Vector Database Layer for RAG Bots (Emtinan)

# Submission

## Interim: Due Wednesday 08 May 20:00 UTC

1. A Pdf report with an overview of your understanding of the key LLM tools and APIs
   1. Review of OpenAI technologies
   2. Review of LangChain components
   3. Review of Vector Databases
   4. Review of LLM based AI applications e.g Retrieval Augmented Generation s and Agents
   5. Review of Redash server, worker, and scheduler. Comment on the use/need for asynchronous computing in chatbot development.
2. A PDF report providing a concise and comprehensive analysis of the SQL query used to address **Task 2**, focusing on its efficiency and performance under scaling conditions. Comment on how long (time) will it take your query if the data size grows by 10x, 1000x, 100000x, 1000000x.
3. Github link submission that demonstrates:
4. Work in progress for Redash chat add-on frontend
5. Work in progress for Redash chat add-on backend
6. Work in progress for prompts to help generate SQL using OpenAI APIs
7. Work in progress for
   1. database schema design
   2. data loading to database
   3. Dockerfile and docker-compose based Redas installation
   4. Celery

### 

### Feedback

You may not receive detailed comments on your interim submission but will receive a grade.

## Interim 2: Due Saturday 11 May 20:00 UTC

1. Pdf document (to be published) or a published Blog link detailing the process followed. This should include:
2. The business objective of the project
3. The project design
4. The tech-stack used
5. The methodologies followed
6. The challenges faced
7. Github link submission that includes the following
   1. Work in progress Redash Chat Add-on frontend code
   2. Work in progress Redash Chat Add-on backend code
   3. Database schema design
   4. Data loading to database
   5. Installation scripts (Dockerfile, docker-compose, bash/python/Makefile scripts to automate installation) and Readme explaining how to install and use
   6. Well demonstrated CI (frequent commits, multiple branches with good names, unit test and linting github actions) and CD (github actions to build docker images from PR to dev and prod branches)

## Final: Due Tuesday 14 May 20:00 UTC

1. Pdf document (to be published) or a published Blog link detailing the process followed. This should include:
2. The business objective of the project
3. The project design
4. The tech-stack used
5. The methodologies followed
6. The challenges faced
7. The results obtained
8. The lessons learned
9. Limitations and future plans
10. Github link submission that includes the following
11. Redash Chat Add-on frontend code
12. Redash Chat Add-on backend code
13. Installation scripts (Dockerfile, docker-compose, bash/python/Makefile scripts to automate installation) and Readme explaining how to install and use
14. Well demonstrated CI (frequent commits, multiple branches with good names, unit test and linting github actions) and CD (github actions to build docker images from PR to dev and prod branches)
15. Demonstration of coding best practices (Following python-openai or langchain code base coding style)

### Feedback

You will receive comments/feedback in addition to a grade.

# References

Key References

* [Redash ApI Usage](https://www.restack.io/docs/redash-knowledge-redash-api-usage-guide)
* [Prompt engineering](https://platform.openai.com/docs/guides/prompt-engineering/prompt-engineering) : Openai Prompt Engineering Guide
* [promptingguide.ai:](https://www.promptingguide.ai/) A prompt engineering guide that demonstrates many techniques.
* [learnprompting.org](https://learnprompting.org/docs/intro): An introductory course to prompt engineering.
* [YouTube API reference](https://developers.google.com/youtube/v3/getting-started)
* [How to extract YouTube data](https://www.analyticssteps.com/blogs/how-extract-analyze-youtube-data-using-youtube-api)
* [YouTube data usage](https://www.python-engineer.com/posts/youtube-data-api-01/)