CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

Development of a Multi-Agent Data Analytics System for Real-Time Social Media Analysis

by

Badri Narayanan S(2348507)

Christina J. Thattil (2348511)

Reeve R. Mathew (2348573)

Under the guidance of

Dr. Thirunavukkarasu

Project Report submitted in partial fulfillment of the requirements of trimester IV
MSAIM, CHRIST (Deemed to be University)

August 2024

TABLE OF CONTENTS

1. Introduction

In the era of big data, the ability to analyze and derive insights from real-world data has become crucial. Social media platforms are goldmines of information, constantly producing a wealth of data that reflects trends, opinions, and behaviors. But how do we harness this vast data effectively? The answer lies in developing a robust, multi-agent data analytics system capable of scraping, processing, and analyzing social media data in real-time, using cutting-edge tools like crewAI, LangChain, and ChatGroq.

In the contemporary landscape of big data, the ability to analyze and derive actionable insights from vast and complex datasets has become essential for organizations and individuals alike. Among the most abundant sources of real-time data are social media platforms, which serve as dynamic repositories of trends, opinions, and behavioral insights.

However, the challenge lies in effectively harnessing this vast and heterogeneous data. Traditional data processing methods often fall short when faced with the volume, velocity, and variety of social media data.

To address these challenges, a robust, multi-agent data analytics system is required one that can efficiently scrape, process, and analyze social media data in real-time.

2. Problem Statement and Objectives

2.1 Problem Statement

1. The primary challenge addressed in this project is the effective harnessing of real-time data from social media platforms for meaningful analysis.
2. Social media platforms like Reddit are rich sources of information, but the vast, unstructured nature of the data makes it difficult to analyze without a structured approach.
3. Understanding public sentiment and monitoring trends in real-time through social media data analysis is crucial for making informed decisions.
4. Existing approaches lack automation and real-time processing capabilities, posing significant challenges in leveraging social media data effectively.

2.2 Objectives

1. **Developed a Multi-Agent Data Analytics System**: Created a system that can efficiently scrape, preprocess, and analyze data from social media platforms in real-time.
2. **Implemented Multiple LLM Models**: Integrated and evaluated the performance of various LLM models on the scraped data.
3. **Compared Model Performance**: Analyzed and compared the performance of the LLM models to determine their effectiveness in processing and analyzing social media data.
4. **Documented the Process and Findings**: Provided detailed documentation of the system's development, including technical insights and recommendations for future work.

3. System Design

3.1 Multi-Agent Setup

The system is designed using a multi-agent architecture, where each agent has a specific role to ensure modularity and scalability. This approach allows the system to efficiently handle the complex tasks involved in data scraping, preprocessing, and analysis.

3.1.1 Scraping Agents

These agents are responsible for gathering data from social media platforms like Reddit. They use APIs and web scraping techniques to collect relevant data in real-time, ensuring that the data is up-to-date and reflects current trends and opinions.

3.1.2 Preprocessing Agents

Once the data is scraped, preprocessing agents clean and prepare it for analysis. This step involves filtering out irrelevant information, handling missing data, and transforming the data into a format suitable for machine learning models.

3.1.3 Analyzer Agents

Analyzer agents implement multiple machine learning models to analyze the preprocessed data. These agents are designed to evaluate the performance of the models using various metrics, enabling a comprehensive comparison of their effectiveness.

3.2 Workflow

The workflow of the system is designed to be linear and efficient, with each step handled by specialized agents:

1. **Data Scraping**: Scraping agents collect data from social media platforms like Reddit.
2. **Data Preprocessing**: Preprocessing agents clean and prepare the data.
3. **Model Implementation and Analysis**: Analyzer agents implement and evaluate multiple LLM models on the preprocessed data.
4. **Result Compilation**: The system compiles the results, including performance metrics such as accuracy, precision, recall, and F1 score.

3.3 Tech Stack

The project leverages the following tools and libraries to implement the multi-agent system:

1. **crewAI**: A framework for creating and managing AI agents, providing the backbone for the multi-agent architecture.
2. **LangChain**: A library that helps manage complex workflows involving LLMs, facilitating seamless integration and execution of multiple models.
3. **ChatGroq**: An LLM inference engine used to evaluate the performance of different models.
4. **Python**: The primary programming language used for all coding tasks.
5. **Serper Development Tools**: Integrated with crewAI for real-time data collection from Google indexes and Wikipedia articles.

4. Problem-Solving Approach

4.1 Data Collection

Data collection is the first critical step in the system's workflow. Scraping agents use APIs and web scraping techniques to gather data from platforms like Reddit. This data is unstructured and requires significant preprocessing before it can be analyzed.

4.2 Data Preprocessing

The scraped data is then passed to preprocessing agents, which clean and prepare the data for analysis. This step involves several key processes:

1. **Data Cleaning**: Removing irrelevant or duplicate data, handling missing values, and filtering out noise.
2. **Data Transformation**: Converting the data into a format suitable for analysis, such as vectorization for text data.
3. **Feature Engineering**: Extracting and selecting relevant features that can enhance the performance of the models.

4.3 Model Implementation and Analysis

Once the data is preprocessed, analyzer agents implement multiple LLM models to analyze the data. These models are evaluated using various metrics to determine their effectiveness. The models implemented in this project include:

1. Llama3.1–70b-versatile
2. Llama3–70b-8192
3. Llama3–8b-8192
4. Gemma2–9b-it
5. Mixtral-8x7b-32768

4.4 Model Comparison

A key aspect of this project is the comparison of different LLM models. Using the ChatGroq inference engine, each model was evaluated on the same dataset to ensure a fair comparison. The performance metrics used for evaluation includes accuracy, recall, precision, f1-score, perplexity, ROC AUC etc. The model evaluation will be done by the separate agent and it will give the results in the markdown format.

5. Results and Recommendations

5.1 Results

The performance of the LLM models was evaluated using the metrics mentioned above. Among the models tested, the **Llama3.1–70b-versatile** model demonstrated the best overall performance, particularly excelling in sentiment analysis tasks.

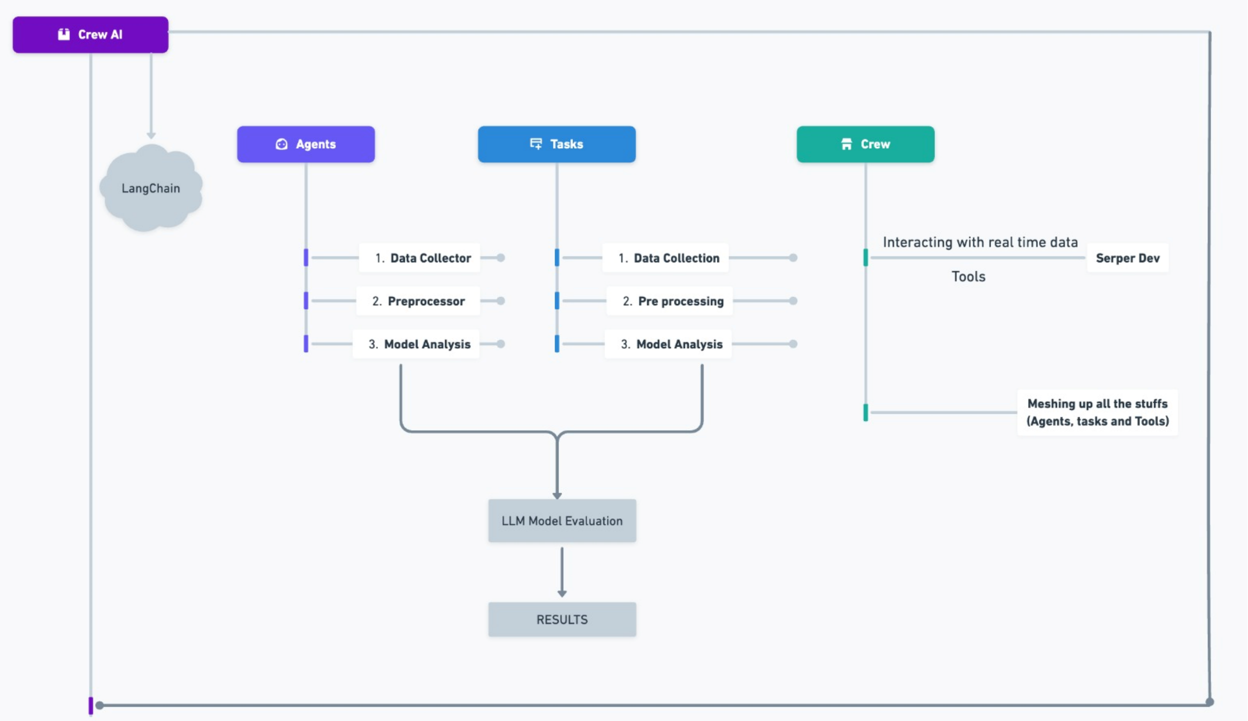
The results for this model are summarized below:

Metric	Value
Sentiment Analysis Accuracy	92.1%
Precision	0.91
Recall	0.93
F1 Score	0.92
Perplexity	12.5

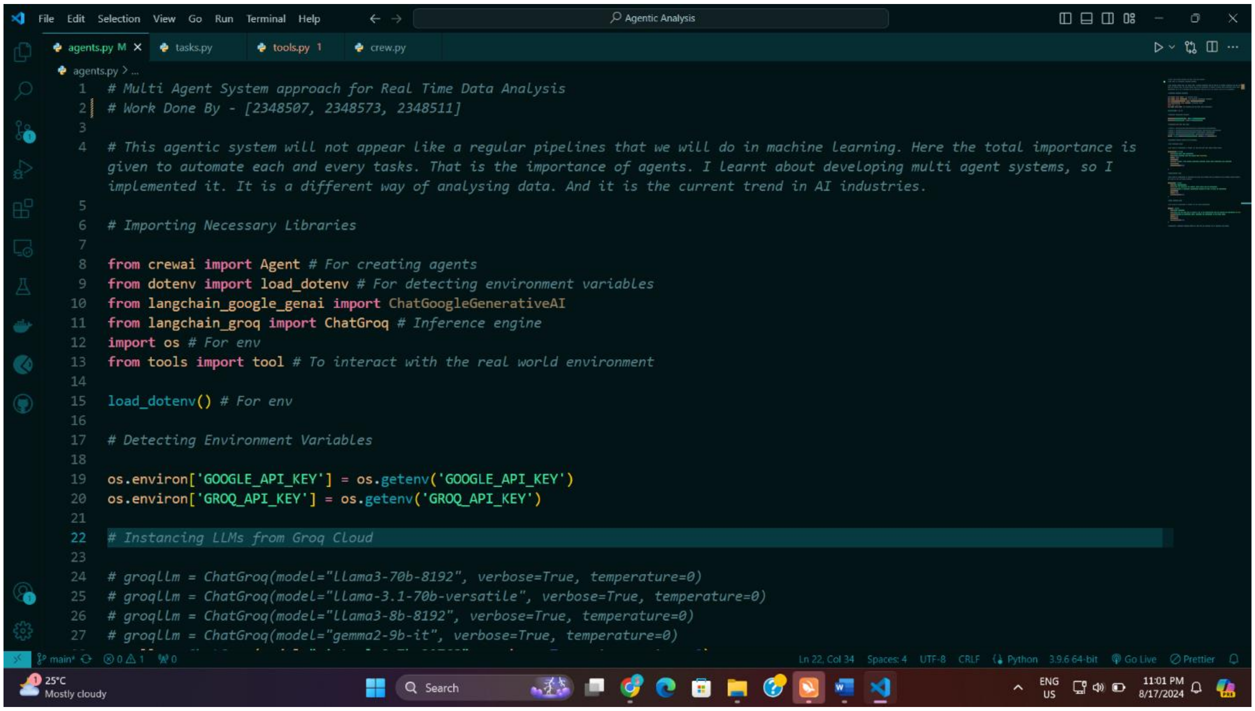
5.2 Recommendations

Based on the results, the **Llama3.1–70b-versatile** model is recommended for tasks involving sentiment analysis. Its high accuracy and balanced precision-recall metrics make it particularly suitable for analyzing social media data, where understanding sentiment is crucial. For future work, it would be beneficial to explore the use of ensemble methods to combine the strengths of multiple models, potentially improving performance further.

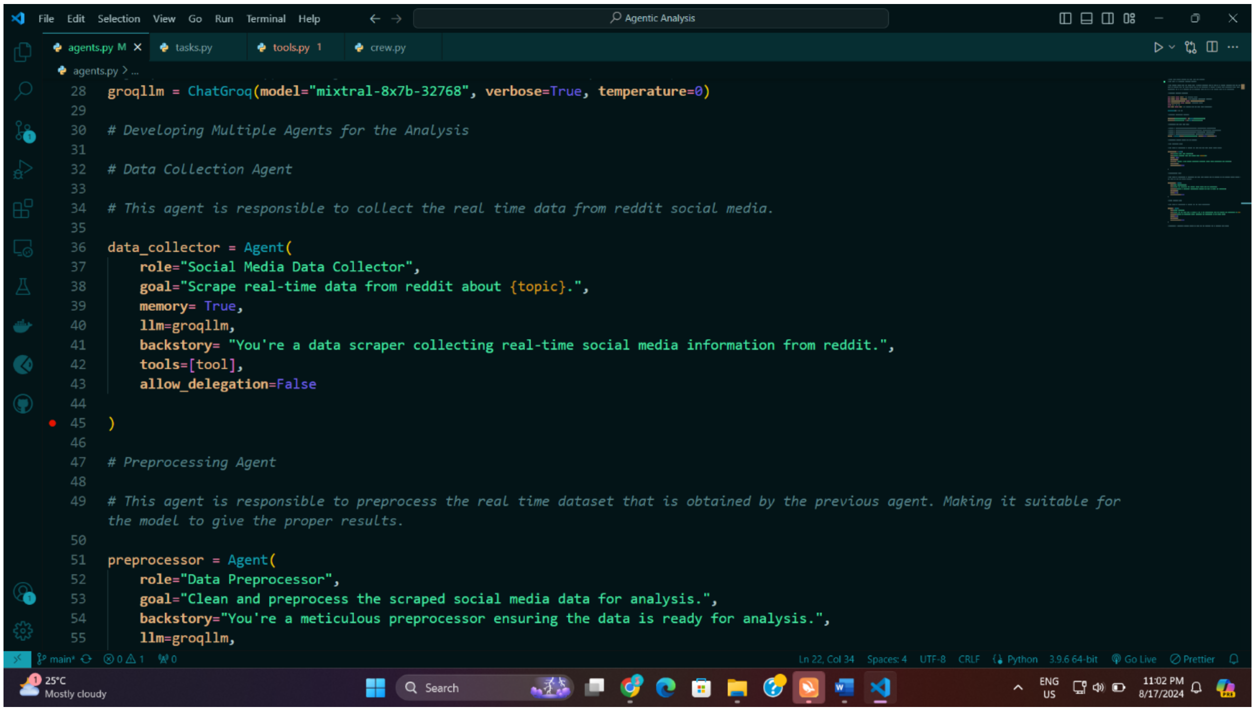
6. Flowchart



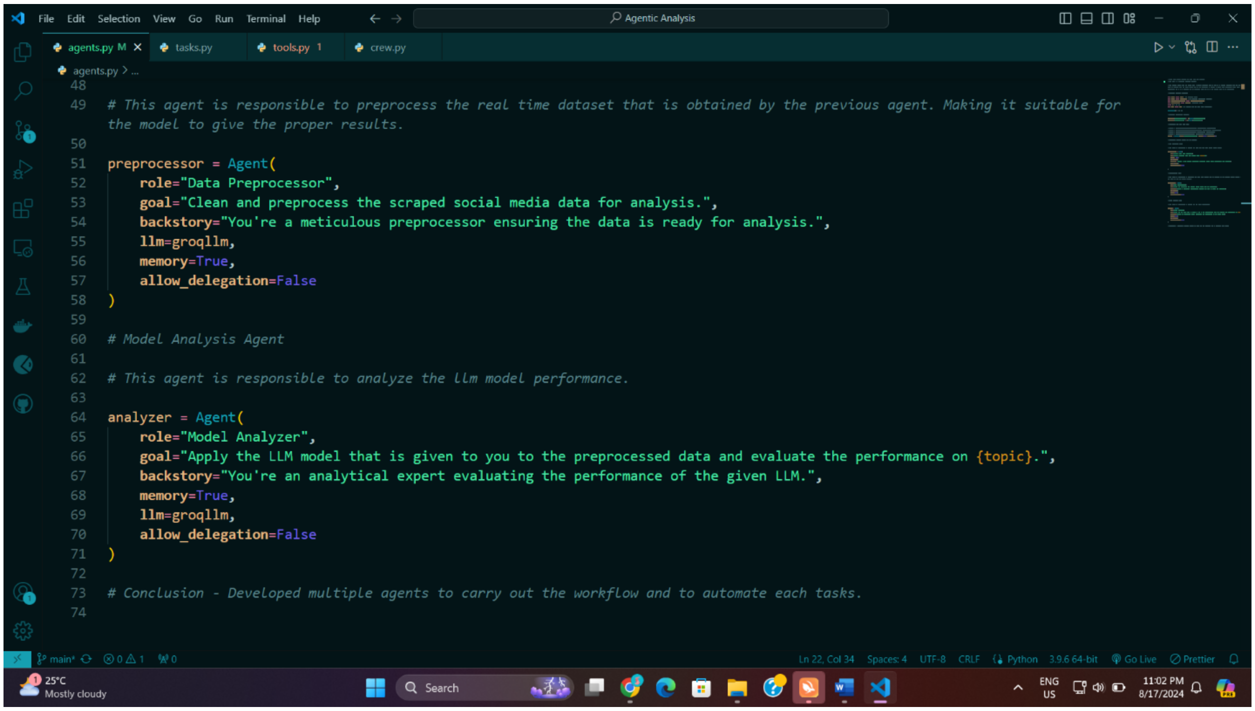
7. Screenshots



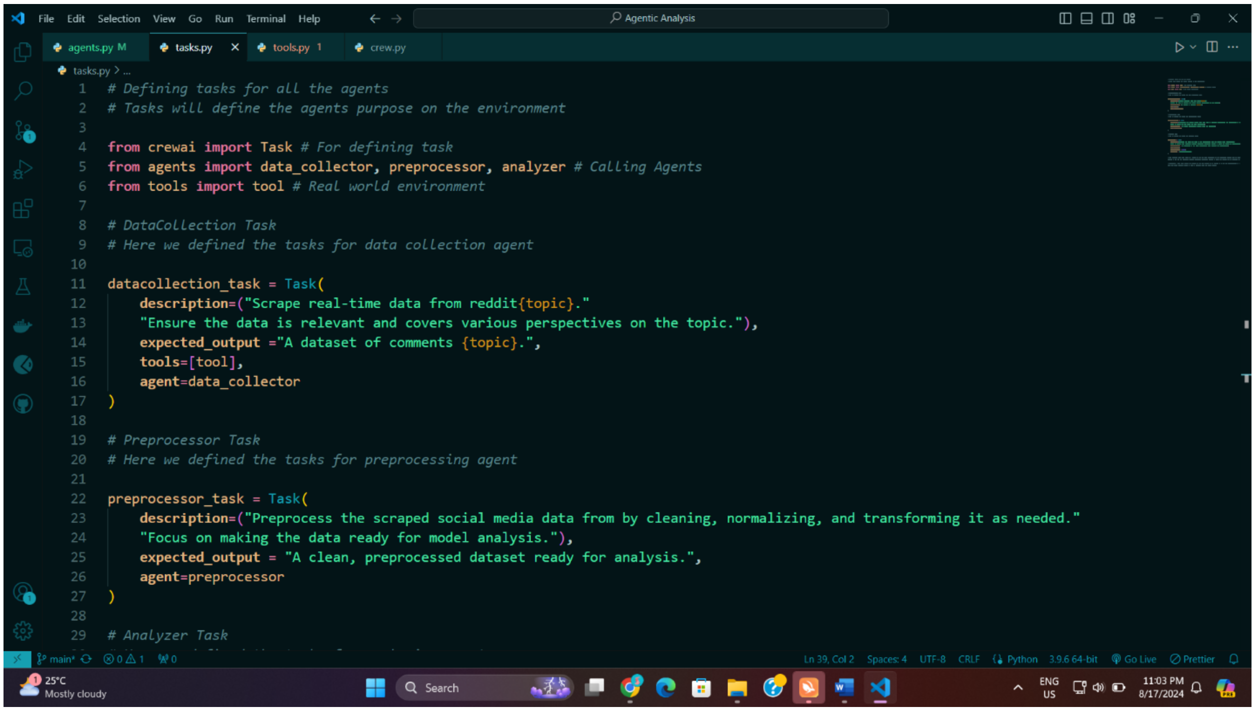
```python
# Multi Agent System approach for Real Time Data Analysis
# Work Done By - [2348507, 2348573, 2348511]

# This agentic system will not appear like a regular pipelines that we will do in machine learning. Here the total importance is given to automate each and every tasks. That is the importance of agents. I learnt about developing multi agent systems, so I implemented it. It is a different way of analysing data. And it is the current trend in AI industries.

# Importing Necessary Libraries

from crewai import Agent # For creating agents
from dotenv import load_dotenv # For detecting environment variables
from langchain_google_genai import ChatGoogleGenerativeAI
from langchain_groq import ChatGroq # Inference engine
import os # For env
from tools import tool # To interact with the real world environment

load_dotenv() # For env

# Detecting Environment Variables

os.environ['GOOGLE_API_KEY'] = os.getenv('GOOGLE_API_KEY')
os.environ['GROQ_API_KEY'] = os.getenv('GROQ_API_KEY')

# Instancing LLMs from Groq Cloud

# groqllm = ChatGroq(model="llama3-70b-8192", verbose=True, temperature=0)
# groqllm = ChatGroq(model="llama-3.1-70b-versatile", verbose=True, temperature=0)
# groqllm = ChatGroq(model="llama3-8b-8192", verbose=True, temperature=0)
# groqllm = ChatGroq(model="gemma2-9b-it", verbose=True, temperature=0)
```



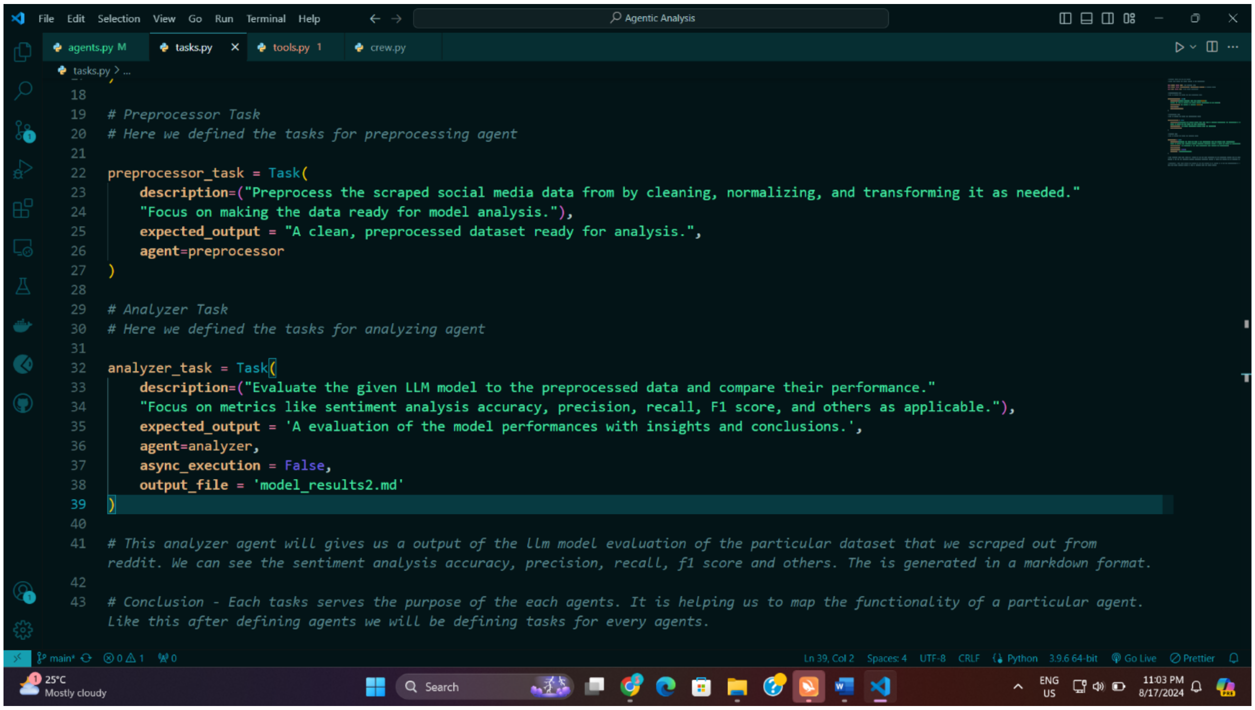
```python
groqllm = ChatGroq(model="mixtral-8x7b-32768", verbose=True, temperature=0)

# Developing Multiple Agents for the Analysis

# Data Collection Agent

# This agent is responsible to collect the real time data from reddit social media.

data_collector = Agent(
    role="Social Media Data Collector",
    goal="Scrape real-time data from reddit about {topic}.",
    memory= True,
    llm=groqllm,
    backstory= "You're a data scraper collecting real-time social media information from reddit.",
    tools=[tool],
    allow_delegation=False

)

# Preprocessing Agent

# This agent is responsible to preprocess the real time dataset that is obtained by the previous agent. Making it suitable for the model to give the proper results.

preprocessor = Agent(
    role="Data Preprocessor",
    goal="Clean and preprocess the scraped social media data for analysis.",
    backstory="You're a meticulous preprocessor ensuring the data is ready for analysis.",
    llm=groqllm,
```



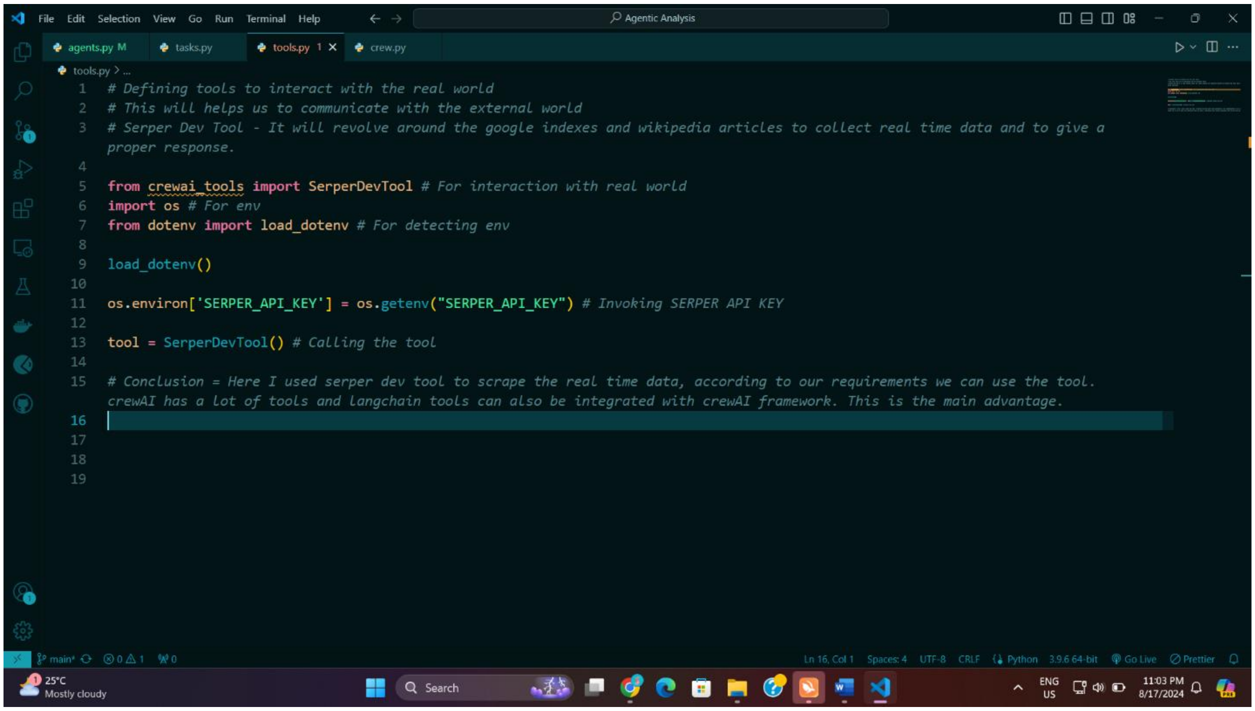
```python

# This agent is responsible to preprocess the real time dataset that is obtained by the previous agent. Making it suitable for the model to give the proper results.

preprocessor = Agent(
    role="Data Preprocessor",
    goal="Clean and preprocess the scraped social media data for analysis.",
    backstory="You're a meticulous preprocessor ensuring the data is ready for analysis.",
    llm=groqllm,
    memory=True,
    allow_delegation=False
)

# Model Analysis Agent

# This agent is responsible to analyze the llm model performance.

analyzer = Agent(
    role="Model Analyzer",
    goal="Apply the LLM model that is given to you to the preprocessed data and evaluate the performance on {topic}.",
    backstory="You're an analytical expert evaluating the performance of the given LLM.",
    memory=True,
    llm=groqllm,
    allow_delegation=False
)

# Conclusion - Developed multiple agents to carry out the workflow and to automate each tasks.

```



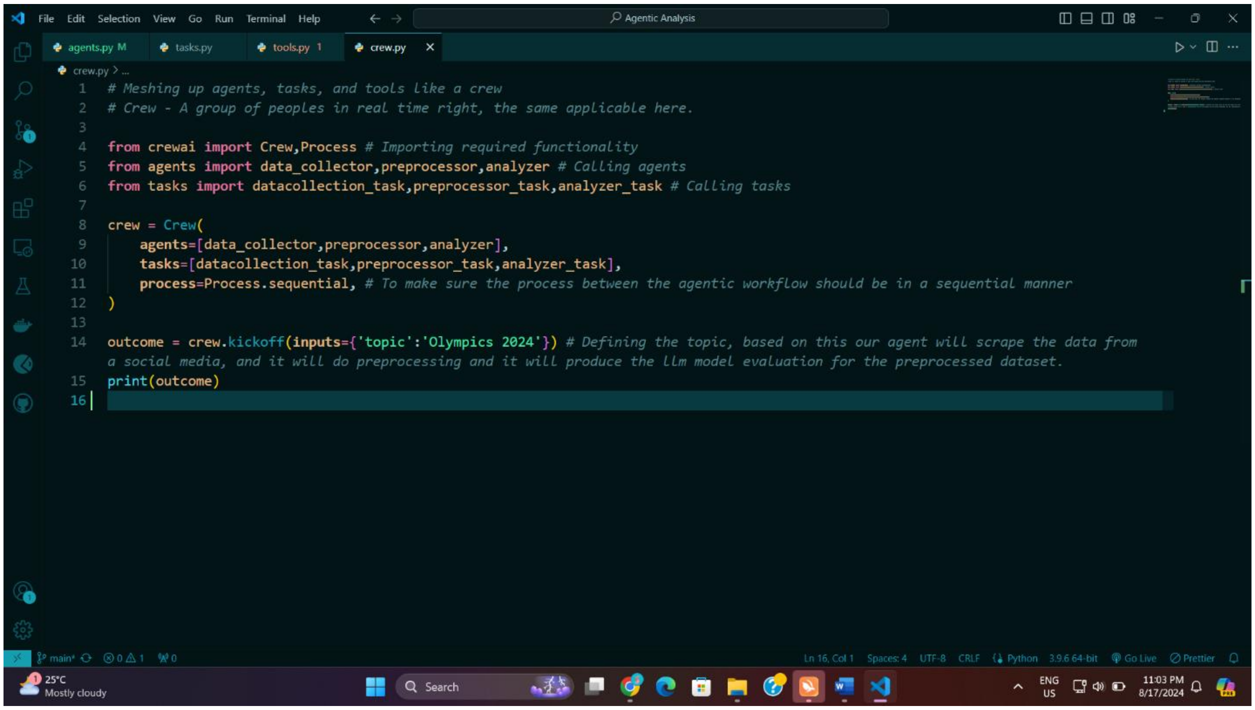
```python
# Defining tasks for all the agents
# Tasks will define the agents purpose on the environment

from crewai import Task # For defining task
from agents import data_collector, preprocessor, analyzer # Calling Agents
from tools import tool # Real world environment

# DataCollection Task
# Here we defined the tasks for data collection agent

datacollection_task = Task(
    description=("Scrape real-time data from reddit{topic}."
    "Ensure the data is relevant and covers various perspectives on the topic."),
    expected_output ="A dataset of comments {topic}.",
    tools=[tool],
    agent=data_collector
)

# Preprocessor Task
# Here we defined the tasks for preprocessing agent

preprocessor_task = Task(
    description=("Preprocess the scraped social media data from by cleaning, normalizing, and transforming it as needed."
    "Focus on making the data ready for model analysis."),
    expected_output = "A clean, preprocessed dataset ready for analysis.",
    agent=preprocessor
)

# Analyzer Task
```



```python

# Preprocessor Task
# Here we defined the tasks for preprocessing agent

preprocessor_task = Task(
    description=("Preprocess the scraped social media data from by cleaning, normalizing, and transforming it as needed."
    "Focus on making the data ready for model analysis."),
    expected_output = "A clean, preprocessed dataset ready for analysis.",
    agent=preprocessor
)

# Analyzer Task
# Here we defined the tasks for analyzing agent

analyzer_task = Task(
    description=("Evaluate the given LLM model to the preprocessed data and compare their performance."
    "Focus on metrics like sentiment analysis accuracy, precision, recall, F1 score, and others as applicable."),
    expected_output = 'A evaluation of the model performances with insights and conclusions.',
    agent=analyzer,
    async_execution = False,
    output_file = 'model_results2.md'
)

# This analyzer agent will gives us a output of the llm model evaluation of the particular dataset that we scraped out from reddit. We can see the sentiment analysis accuracy, precision, recall, f1 score and others. The is generated in a markdown format.

# Conclusion - Each tasks serves the purpose of the each agents. It is helping us to map the functionality of a particular agent. Like this after defining agents we will be defining tasks for every agents.
```



```python
# Defining tools to interact with the real world
# This will helps us to communicate with the external world
# Serper Dev Tool - It will revolve around the google indexes and wikipedia articles to collect real time data and to give a proper response.

from crewai_tools import SerperDevTool # For interaction with real world
import os # For env
from dotenv import load_dotenv # For detecting env

load_dotenv()

os.environ['SERPER_API_KEY'] = os.getenv("SERPER_API_KEY") # Invoking SERPER API KEY

tool = SerperDevTool() # Calling the tool

# Conclusion = Here I used serper dev tool to scrape the real time data, according to our requirements we can use the tool. crewAI has a lot of tools and langchain tools can also be integrated with crewAI framework. This is the main advantage.



```



File Edit Selection View Go Run Terminal Help
Agentic Analysis
agents.py M
tasks.py
tools.py 1
crew.py
crew.py > ...
Ln 16, Col 1 Spaces: 4 UTF-8 CRLF Python 3.9.6 64-bit Go Live Prettier
main*
25°C
Mostly cloudy
Search
ENG
US
11:03 PM
8/17/2024

```python
# Meshing up agents, tasks, and tools like a crew
# Crew - A group of peoples in real time right, the same applicable here.

from crewai import Crew,Process # Importing required functionality
from agents import data_collector,preprocessor,analyzer # Calling agents
from tasks import datacollection_task,preprocessor_task,analyzer_task # Calling tasks

crew = Crew(
    agents=[data_collector,preprocessor,analyzer],
    tasks=[datacollection_task,preprocessor_task,analyzer_task],
    process=Process.sequential, # To make sure the process between the agentic workflow should be in a sequential manner
)

outcome = crew.kickoff(inputs={'topic':'Olympics 2024'}) # Defining the topic, based on this our agent will scrape the data from a social media, and it will do preprocessing and it will produce the llm model evaluation for the preprocessed dataset.
print(outcome)

```

8. Conclusion

8.1 Summary of Findings

1. Successfully developed a Multiagent System (MAS) for automating real-time data analysis tasks.
2. By leveraging crewAI, LangChain, and ChatGroq, the system was able to scrape, preprocess, and analyze data efficiently.
3. Demonstrated efficient handling and analysis of large-scale social media data from Reddit.
4. Highlighted the flexibility and modularity of the MAS, allowing easy adaptation to various tasks.
5. Provided accurate and timely insights, particularly in sentiment analysis.
6. The Llama3.1-70b-versatile model emerged as the best performer, demonstrating the power of advanced LLMs in sentiment analysis tasks. This project highlights the effectiveness of agentic approaches in AI, paving the way for future developments in real-time data analytics.

8.2 Future Work

1. Extend the system to support additional data sources, such as Twitter, Facebook, or news websites.
2. Enhance agent collaboration for improved efficiency and accuracy.
3. Integrate more advanced machine learning models and expand the range of analysis tasks, such as trend prediction or anomaly detection.

8.3 Final Thoughts

1. The MAS represents a significant advancement in automating real-time data analysis.
2. Reduces manual intervention and enables continuous processing of data.
3. Offers wide-ranging applications across industries, including social media monitoring and financial analysis.
4. As demand for real-time insights grows, MAS adoption is expected to increase, providing a powerful tool for organizations in a data-driven world.

References

1. **Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020).** Language models are few-shot learners. *Advances in Neural Information Processing Systems, 33*, 1877-1901.
2. **Henderson, P., Hu, J., Romoff, J., Brunskill, E., Jurafsky, D., & Pineau, J. (2018).** Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research, 21*(248), 1-43.
3. **Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017).** Attention is all you need. *Advances in Neural Information Processing Systems, 30*, 5998-6008.
4. **Zhang, Y., & Wallace, B. C. (2015).** A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 253-263.