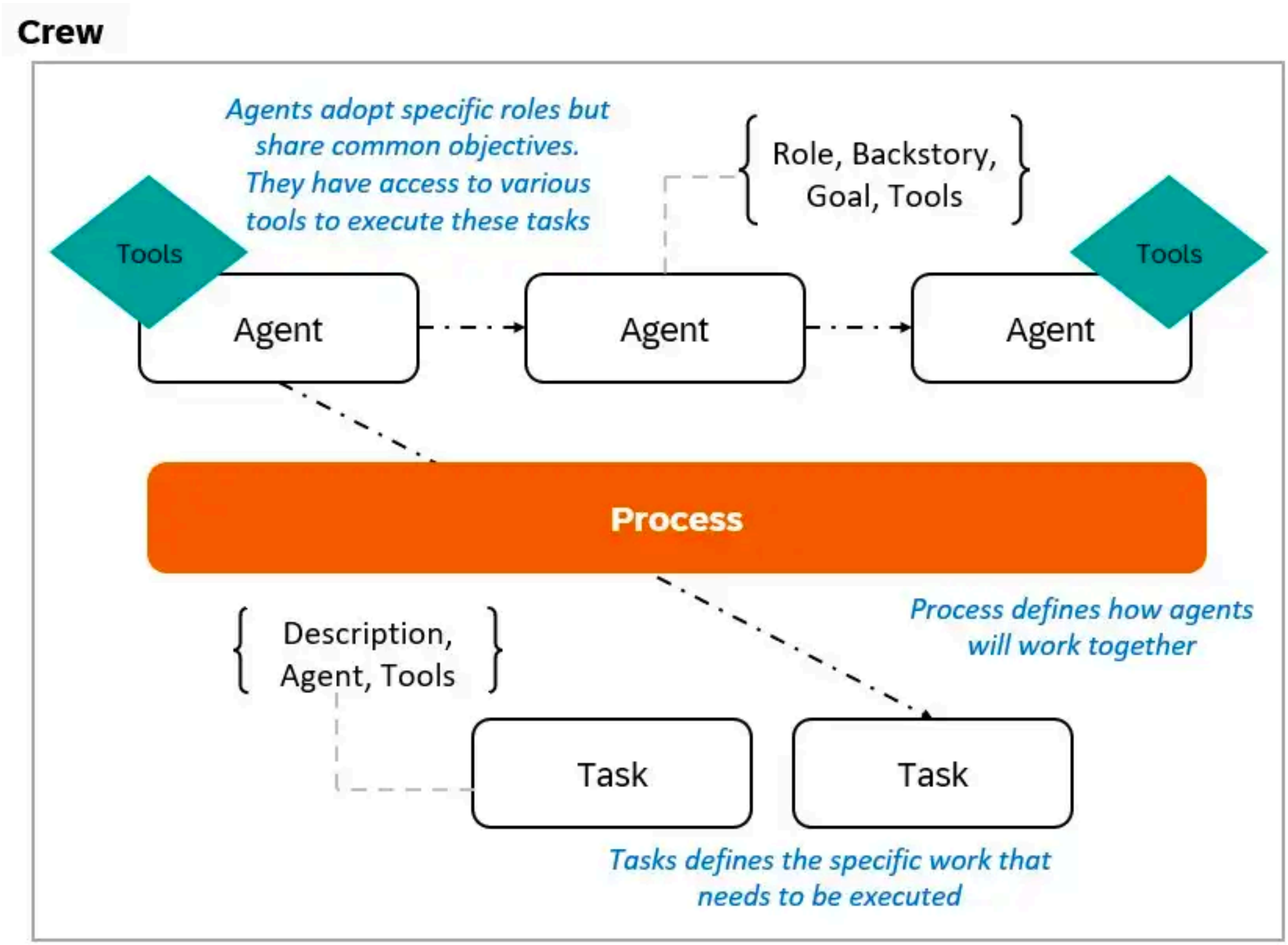
Analytics Vidhya

Day 13 of Mastering AI Agents

# Build AI Agents with CrewAI



# Problem Description

Imagine you are running an education counselling company and want to suggest the best courses for your students based on their degrees, academic goals, hobbies, and computer skills. How do you decide which courses to recommend to each student? For example, it would be more logical to suggest a course on climate change to a student studying Environmental Science rather than one majoring in Computer Science.

The student's profile dataset looks like the following:

| | Academic Goals | Major | Hobbies | Computer Skills | Interest in Languages | GPA |
|---|---|---|---|---|---|---|
| 0 | To become a software engineer | Computer Science | Gaming | Advanced | Spanish | 3.7 |
| 1 | To study environmental science | Environmental Science | Hiking | Intermediate | French | 3.5 |
| 2 | To pursue a career in medicine | Pre-Med | Playing the piano | Advanced | Spanish | 3.9 |
| 3 | To major in psychology | Psychology | Reading | Intermediate | German | 3.6 |
| 4 | To work in international relations | Political Science | Traveling | Basic | Mandarin | 3.8 |
| 5 | To become a teacher | Education | Painting | Advanced | Spanish | 3.4 |
| 6 | To study literature | English Literature | Writing | Intermediate | French | 3.9 |
| 7 | To pursue a career in business | Business Administration | Playing soccer | Basic | Mandarin | 3.5 |
| 8 | To become a biologist | Biology | Photography | Advanced | German | 3.7 |
| 9 | To work in data analysis | Statistics | Cooking | Intermediate | Japanese | 3.6 |

The possible list of courses that should be used for recommendation to the students could be the following:

```
"Introduction to Computer Science" - Offered by Harvard University on edX
"Biology: Life on Earth" - Offered by Coursera
"Introduction to Psychology" - Offered by Yale University on Coursera
"Environmental Science" - Offered by University of Leeds on FutureLearn
"Introduction to Literature" - Offered by MIT on edX
"Medical Terminology" - Offered by University of Pittsburgh on Coursera
"Data Science and Machine Learning" - Offered by Stanford University on Coursera
"Cell Biology" - Offered by Massachusetts Institute of Technology on edX
"Positive Psychology" - Offered by University of North Carolina at Chapel Hill on Coursera
"Environmental Law and Policy" - Offered by Vermont Law School on Coursera
"Programming for Everybody (Getting Started with Python)" - Offered by University of Michigan on Coursera
"Anatomy: Human Neuroanatomy" - Offered by University of Michigan on Coursera
"Introduction to Cognitive Psychology" - Offered by Duke University on Coursera
"Climate Change and Health: From Science to Action" - Offered by Harvard University on edX
"English for Science, Technology, Engineering, and Mathematics" - Offered by University of Pennsylvania on Coursera
"An Introduction to American Law" - Offered by University of Pennsylvania on Coursera
"Introduction to Chemistry: Reactions and Ratios" - Offered by Duke University on Coursera
"Epidemiology: The Basic Science of Public Health" - Offered by University of North Carolina at Chapel Hill on Coursera
"Computer Science: Programming with a Purpose" - Offered by Princeton University on Coursera
"Introduction to Statistics and Data Analysis" - Offered by Rice University on Coursera
"Genes and the Human Condition (From Behavior to Biotechnology)" - Offered by University of Maryland on Coursera
"Ethics, Technology, and the Future of Medicine" - Offered by Georgetown University on edX
"Fundamentals of Immunology" - Offered by Harvard University
```

Our first Crew consists of the following AI agents, with the help of which we will first generate the top 3 suitable courses for each of the students in our dataset –

- **Chief Recommendation Director**: AI agent that gets the main task and oversees work of others
- **Student Profiler**: AI agent to understand the profiles of the different students
- **Course Specialist**: AI agent that can help in linking appropriate courses to each student

# Python Implementation

Install & Import Necessary Libraries

Let us first start by installing the needed Python libraries –

```
!pip install crewai
!pip install langchain_openai
```

Next, we import all the required Python libraries:

```
import os
import re
import json
import pandas as pd
from tqdm import tqdm
from crewai import Agent, Task, Crew, Process
from langchain_openai import ChatOpenAI
from textwrap import dedent
```

## Define the LLM model with the API key

We then define the LLM model to be used and the respective API key. We can check with any LLM model of our choice. Here, we have utilized gpt-3.5-turbo-16k.

```
import os
openai_api_key = ''
os.environ['OPENAI_API_KEY'] = openai_api_key
llm = ChatOpenAI(
    model="gpt-3.5-turbo-16k",
    temperature=0.1,
    max_tokens=8000
)
```

## Define Datasets To be Used

Next, we define the csv for the student profile data and also the list of products/courses to be recommended to the students:

```
csv = '''Academic Goals, Major, Hobbies, Computer Skills, Interest in Languages, GPA
To become a software engineer, Computer Science, Gaming, Advanced, Spanish, 3.7
To study environmental science, Environmental Science, Hiking, Intermediate, French, 3.5
To pursue a career in medicine, Pre-Med, Playing the piano, Advanced, Spanish, 3.9
To major in psychology, Psychology, Reading, Intermediate, German, 3.6
To work in international relations, Political Science, Traveling, Basic, Mandarin, 3.8
To become a teacher, Education, Painting, Advanced, Spanish, 3.4
To study literature, English Literature, Writing, Intermediate, French, 3.9
To pursue a career in business, Business Administration, Playing soccer, Basic, Mandarin, 3.5
To become a biologist, Biology, Photography, Advanced, German, 3.7
To work in data analysis, Statistics, Cooking, Intermediate, Japanese, 3.6
'''
#creating the df_customers dataframe
from io import StringIO
# Convert string into StringIO
csvStringIO = StringIO(csv)
df_customers = pd.read_csv(csvStringIO, sep=",")


products = '''
"Introduction to Computer Science" - Offered by Harvard University on edX
"Biology: Life on Earth" - Offered by Coursera
"Introduction to Psychology" - Offered by Yale University on Coursera
"Environmental Science" - Offered by University of Leeds on FutureLearn
"Introduction to Literature" - Offered by MIT on edX
"Medical Terminology" - Offered by University of Pittsburgh on Coursera
"Data Science and Machine Learning" - Offered by Stanford University on Coursera
"Cell Biology" - Offered by Massachusetts Institute of Technology on edX
"Positive Psychology" - Offered by University of North Carolina at Chapel Hill on Coursera
"Environmental Law and Policy" - Offered by Vermont Law School on Coursera
"Programming for Everybody (Getting Started with Python)" - Offered by University of Michigan on Coursera
"Anatomy: Human Neuroanatomy" - Offered by University of Michigan on Coursera
"Introduction to Cognitive Psychology" - Offered by Duke University on Coursera
"Climate Change and Health: From Science to Action" - Offered by Harvard University on edX
"English for Science, Technology, Engineering, and Mathematics" - Offered by University of Pennsylvania on Coursera
"An Introduction to American Law" - Offered by University of Pennsylvania on Coursera
"Introduction to Chemistry: Reactions and Ratios" - Offered by Duke University on Coursera
"Epidemiology: The Basic Science of Public Health" - Offered by University of North Carolina at Chapel Hill on Coursera
"Computer Science: Programming with a Purpose" - Offered by Princeton University on Coursera
"Introduction to Statistics and Data Analysis" - Offered by Rice University on Coursera
"Genes and the Human Condition (From Behavior to Biotechnology)" - Offered by University of Maryland on Coursera
"Ethics, Technology, and the Future of Medicine" - Offered by Georgetown University on edX
"Fundamentals of Immunology" - Offered by Harvard University
'''
```

## Define the LLM model with the API key

We then define the LLM model to be used and the respective API key. We can check with any LLM model of our choice. Here, we have utilized gpt-3.5-turbo-16k.

```
import os
openai_api_key = ''
os.environ['OPENAI_API_KEY'] = openai_api_key
llm = ChatOpenAI(
    model="gpt-3.5-turbo-16k",
    temperature=0.1,
    max_tokens=8000
)
```

## Define Datasets To be Used

Next, we define the csv for the student profile data and also the list of products/courses to be recommended to the students:

```
csv = '''Academic Goals, Major, Hobbies, Computer Skills, Interest in Languages, GPA
To become a software engineer, Computer Science, Gaming, Advanced, Spanish, 3.7
To study environmental science, Environmental Science, Hiking, Intermediate, French, 3.5
To pursue a career in medicine, Pre-Med, Playing the piano, Advanced, Spanish, 3.9
To major in psychology, Psychology, Reading, Intermediate, German, 3.6
To work in international relations, Political Science, Traveling, Basic, Mandarin, 3.8
To become a teacher, Education, Painting, Advanced, Spanish, 3.4
To study literature, English Literature, Writing, Intermediate, French, 3.9
To pursue a career in business, Business Administration, Playing soccer, Basic, Mandarin, 3.5
To become a biologist, Biology, Photography, Advanced, German, 3.7
To work in data analysis, Statistics, Cooking, Intermediate, Japanese, 3.6
'''
#creating the df_customers dataframe
from io import StringIO
# Convert string into StringIO
csvStringIO = StringIO(csv)
df_customers = pd.read_csv(csvStringIO, sep=",")


products = '''
"Introduction to Computer Science" - Offered by Harvard University on edX
"Biology: Life on Earth" - Offered by Coursera
"Introduction to Psychology" - Offered by Yale University on Coursera
"Environmental Science" - Offered by University of Leeds on FutureLearn
"Introduction to Literature" - Offered by MIT on edX
"Medical Terminology" - Offered by University of Pittsburgh on Coursera
"Data Science and Machine Learning" - Offered by Stanford University on Coursera
"Cell Biology" - Offered by Massachusetts Institute of Technology on edX
"Positive Psychology" - Offered by University of North Carolina at Chapel Hill on Coursera
"Environmental Law and Policy" - Offered by Vermont Law School on Coursera
"Programming for Everybody (Getting Started with Python)" - Offered by University of Michigan on Coursera
"Anatomy: Human Neuroanatomy" - Offered by University of Michigan on Coursera
"Introduction to Cognitive Psychology" - Offered by Duke University on Coursera
"Climate Change and Health: From Science to Action" - Offered by Harvard University on edX
"English for Science, Technology, Engineering, and Mathematics" - Offered by University of Pennsylvania on Coursera
"An Introduction to American Law" - Offered by University of Pennsylvania on Coursera
"Introduction to Chemistry: Reactions and Ratios" - Offered by Duke University on Coursera
"Epidemiology: The Basic Science of Public Health" - Offered by University of North Carolina at Chapel Hill on Coursera
"Computer Science: Programming with a Purpose" - Offered by Princeton University on Coursera
"Introduction to Statistics and Data Analysis" - Offered by Rice University on Coursera
"Genes and the Human Condition (From Behavior to Biotechnology)" - Offered by University of Maryland on Coursera
"Ethics, Technology, and the Future of Medicine" - Offered by Georgetown University on edX
"Fundamentals of Immunology" - Offered by Harvard University
'''
```

## Define Agents To be Used

Post this, we begin defining the different AI agents for our first crew. Each AI agent needs a role, goal and backstory, as shown below in the code block:

```
student_profiler = Agent(
  role='student_profiler',
  goal='''From limited data, you logically deduct conclusions about students.''',
  backstory='You are an expert psychologist with decades of experience.',
  llm = llm,allow_delegation=False,verbose=True)



course_specialist = Agent(
     role='course specialist',
     goal='''Match the suitable course to the students''',
     backstory='You have exceptional knowledge of the courses and can say how valuable they are to a student.',
     llm = llm,allow_delegation=False,verbose=True)


Chief_Recommendation_Director = Agent(
     role="Chief Recomeendation Director",
     goal=dedent("""\Oversee the work done by your team to make sure it's the best
        possible and aligned with the course's goals, review, approve,
        ask clarifying question or delegate follow up work if necessary to make
        decisions"""),
     backstory=dedent("""\You're the Chief Promotion Officer of a large EDtech company. You're launching a
personalized ad campaign,
          trying to make sure your team is crafting the best possible
      content for the customer."""),
     llm = llm,tools=[],allow_delegation=False, verbose=True)
```

The second Crew will be used to generate a recommendation text for the courses recommended to a student.

```
campaign_agent = Agent(
     role="campaign_agent",
     goal=dedent("""\Develop compelling and innovative content
        for ad campaigns, with a focus customer specific ad copies."""),
     backstory=dedent("""\As a Creative Content Creator at a top-tier
           digital marketing agency, you excel in crafting advertisements
           that resonate with potential customers.
           Your expertise lies in turning marketing strategies
           into engaging stories that capture
           attention and inspire buying action."""),
     llm = llm,allow_delegation=False, verbose=True)#import csv
```

For more information, kindly visit this **article**

Advanced AI Agents Generative AI Generative AI Application

## AI Agents with CrewAI: Transforming Edtech Solutions

AI agents with CrewAI collaborate to deliver effective and personalized course recommendations, enhancing the Edtech experience.

*Nibedita Dutta* 18 Mar, 2025