I. Introduction to AI Agents and n8n

1. **What is an AI Agent?**

- An AI agent is a **system that perceives its environment, processes information, and takes action to achieve a goal**, often using machine learning or rule-based logic.
- It senses its surroundings, processes the data, and makes decisions to accomplish a specific objective.
- **Examples**:
  - **ChatGPT**: Takes your input and provides an output to help you achieve a goal, like writing an essay.
  - **Instagram Feed**: Observes your interactions (likes/dislikes) and uses AI to show you more relevant content.
- AI agents are not a new or unique concept; they are common and can be easily built.

2. **What is n8n?**

- n8n is a **workflow automation tool that allows users to automate tasks and integrate various APIs without extensive coding**.
- It enables you to automate any sequence of tasks (workflow) using AI.
- **What is an API?** An API (Application Programming Interface) allows different applications to use their programming interfaces to communicate and build workflows.
- **Example of an n8n Workflow**: An email responder app. It reads incoming emails, classifies them, generates a response using an AI (like OpenAI's API), and sends it back.
- **Benefits of n8n**:
  - **No Coding**: You typically won't need to write code in 99% of cases, even for advanced applications.
  - **Flexible Workflow Creation**: Offers diverse ways to create workflows.
  - **Seamless API Integration**: Easily integrates with various AI APIs such as OpenAI, DeepSeek, and ClaudeAI. (This course focuses on OpenAI's API).

II. Getting Started with n8n

1. **Account Setup**:

- Go to the **n8n website**.
- Click **"Get Started for Free"**.
- n8n is a paid platform, but it offers a **free trial** that includes all features and **does not require a credit card**.
- Fill in your information to sign up and start your 14-day free trial.
- Log in to your n8n account.

2. **n8n User Interface (UI) Overview**:

- The UI displays your existing workflows.
- The left sidebar includes:
  - **Overview**: Shows all your workflows.
  - **Projects**: Where your projects are listed.
  - **Settings/Sign Out**.
- **Templates**: n8n provides pre-built AI agents as templates.
  - **Recommendation**: It's best to learn the basics of building agents from scratch first before relying on templates. This helps you understand the underlying logic.

3. **Creating a Workflow**:

- Click **"Create Workflow"** to open the editor.
- You can move the canvas by clicking the middle mouse button.
- **Nodes**: These are the individual **steps your AI agent will perform**.

4. **Triggering an AI Agent (Adding the First Step)**:

- Click **"Add First Step"**.
- **Trigger Manually**: This option allows you to manually run your AI agent by clicking "Test Workflow".
- **Other Trigger Options**: AI agents can be triggered in many ways:
  - **On Chat Message**: Activates the agent when a message is sent in a chat, similar to interacting with ChatGPT.
  - **On Schedule**: Runs the agent at a specific time or after a fixed interval (e.g., daily, hourly).
  - **On App Event**: Triggers based on events occurring within other applications.
- **Note**: The course teaches settings and options through building projects rather than listing them one by one.

III. Project 1: Inventory Management System

This project builds a chatbot that interacts with a grocery inventory database. It can answer questions about stock, and add or remove items. This is a realistic project that can be sold for freelancing.

1. **Step-by-Step Construction**:

- **Step 1: Add Trigger Node - On Chat Message**.
  - This will trigger the AI agent whenever a chat message is received.
- **Step 2: Add AI Agent Node**.
  - Click the "+" button after the "On Chat" node.
  - Go to **"AI Advanced"** and select **"AI Agent"**.

- Choose **"Tools Agent"** as the agent type. This is perfect for our use case.

- **Step 3: Understanding AI Agent Components**:

  - Every AI agent has three main parts: **Chat Model**, **Memory**, and **Tool**.

- **Step 4: Configure Chat Model (OpenAI)**.

  - Click the "+" next to "Chat Model".

  - **Model Choice**: You can use DeepSeek (good results, cheaper) or **OpenAI Chat Model** (best results, slightly more expensive). We will use OpenAI.

  - **Connect OpenAI Account**:

    - Click **"Create New Credential"**.

    - **Obtain OpenAI API Key**:

      - Go to **platform.openai.com**.

      - Sign up or log in.

      - **Add Credit**: Navigate to **Settings -> Billing -> Payment Methods** and add funds. Testing is very cheap (e.g., $0.04).

      - **Generate API Key**: Go to **API Keys -> Create New Secret Key**. Give it a name (e.g., "n8n Course") and click "Create Key". **Immediately copy this key** as it cannot be viewed again. Store it safely.

      - **Paste the API Key** into n8n.

      - Click "Save". n8n will test the API key.

  - **Select Model**: OpenAI offers various models. For this course, **4o mini** is recommended. It's very affordable ($0.15 per million tokens) and provides good results. 4o offers better results but is more expensive.

  - **Test Chat Model**: You can open the chat and send messages (e.g., "Hello, how are you?"). However, you'll notice it **forgets previous messages**.

- **Step 5: Configure Memory (Window Buffer Memory)**.

  - Click the "+" next to "Memory".

  - Select **"Window Buffer Memory"**.

  - **Context Window Length**: This setting defines how many past messages the AI agent remembers. The default is **5**. Increasing this (e.g., to 100) will increase API costs, while decreasing it (e.g., to 3) can save money for testing.

  - **Test Memory**: Ask "How tall is Burj Khalifa?", then "What about Empire State Building?". With memory, it should now understand the context and provide the height of the Empire State Building.

- **Step 6: Add Tool 1 - Google Sheets (Search Inventory)**.

  - A **tool** is an action your AI agent can perform based on the input it receives.

▪ Click the "+" next to "Tool".

▪ Search for **"Sheets"** and select **"Google Sheets"**.

▪ **Prerequisite**: Create a new Google Sheet (e.g., "n8n Test") with two columns: "Item Name" and "Quantity". Populate it with some grocery items and their stock.

▪ **Connect Google Account**: Connect your Google account (create new credentials if needed, similar to OpenAI).

▪ **Operation**: Select **"Get Rows"** (to retrieve data from the sheet).

▪ **Document**: Select your created Google Sheet (e.g., "n8n Test").

▪ **Sheet Number**: Select "Sheet1".

▪ Go "Back to Canvas".

▪ **Rename** this Google Sheets node to "Search Inventory" for clarity.

▪ **Test**: Ask "How many apples do I have in stock?". Observe the flow: AI agent -> OpenAI model (understands query) -> Google Sheets (retrieves data) -> AI agent -> OpenAI model (formats response) -> Displays answer.

▪ Ask "Is there anything out of stock?". It should identify items with zero quantity.

◦ **Step 7: Add Tool 2 - Google Sheets (Update Inventory)**.

▪ Click the "+" next to "Tool" again.

▪ Search for **"Sheets"** and select **"Google Sheets"**.

▪ **Rename** this node to "Update Inventory".

▪ **Operation**: Select **"Append or Update Row"** (this allows adding new items or changing quantities of existing ones).

▪ **Document/Sheet Number**: Select the same Google Sheet as before.

▪ **Important Settings**:

• **Column to Match On**: Select **"Item Name"**. This tells the AI which column to use to find and match items for updates.

• **Item Name** and **Quantity**: For both, click **"Let the model define this parameter"**. This allows the AI agent to extract the item name and quantity from the user's message itself.

▪ Go "Back to Canvas".

◦ **Step 8: Test the Complete Inventory Agent**:

▪ Ask "How many bananas do I have?" (It should respond with "0 bananas in stock").

▪ Then, instruct it: **"I bought 10 bananas. Add them to my stock."** The agent should automatically update the Google Sheet, and a follow-up query will reflect the new quantity.

▪ Similarly, try **"I sold 5 bottles of milk today. Update inventory."** It should reduce the milk quantity in the sheet.

▪ **AI Agent Intelligence**: The agent can smartly **decide which tool to use** (search or update) based on the user's query, making it a truly intelligent system.

2. **Delivering the AI Agent (Making it Public)**:

◦ **Save your model**: This is crucial as n8n might not auto-save.

◦ **Make Chat Public**: Double-click the "On Chat" trigger node.

◦ Toggle **"Make Chat Publicly Available"** to **ON**. This will generate a URL.

◦ **Activate Workflow**: Go back to the main canvas and toggle the workflow status from "Inactive" to **"Active"**.

◦ Now, the generated URL will provide access to your live AI agent.

◦ **Customize the Chat Interface**:

▪ You can change the **"Initial Message"** (e.g., "Grocery AI Assistant, How can I assist you today?").

▪ **Authentication**: Add email/password for access if needed.

▪ **Hosted Chat vs. Embedded Chat**:

• **Hosted Chat**: The chat appears on a full n8n page.

• **Embedded Chat**: n8n provides instructions (for a web developer) to embed the chat into an existing website.

IV. Project 2: Email Responder System

This project creates an AI agent that automatically reads incoming emails, classifies them into categories (e.g., sponsorship, high priority), and generates a suitable response, which can be sent directly or saved as a draft. This is highly practical for YouTubers and businesses.

1. **Step-by-Step Construction**:

◦ **Step 1: Create a New Workflow** and **Save** it.

◦ **Step 2: Add Trigger Node - Gmail Trigger**.

▪ Click the first "+" button, search for **"Gmail"**, and select **"Gmail Trigger"**.

▪ **Connect Gmail Account**: Click "Create New Credential" and **sign in with your Google account**.

▪ **Run Interval**: This determines how often the agent checks for new emails. **"Every Minute"** provides faster responses but consumes more API calls/credits. **"Every Hour"** is a good balance for cost efficiency.

▪ **Event**: Select **"Message Received"**.

▪ **Simplify**: Turn **OFF**. Keeping it on simplifies the email data and removes the actual email text, which is essential for classification.

▪ **Fetch Test Event**: Click this to retrieve data from your last incoming email. If no recent emails, send a test email to yourself from another account.

▪ Observe the full email data, specifically the **text** (email body) and **subject**.

▪ Go "Back to Canvas".

◦ **Step 3: Add Text Classification Node**.

▪ Click the "+" button after the Gmail trigger node.

▪ Search for **"Text Classification"** and select **"Text Classify"**.

▪ **Attach OpenAI Chat Model**: Click the "+" within the "Text Classify" node to add a model. Select **"OpenAI Chat Model"**.

• Choose **"4o mini"** as the model.

• Connect your OpenAI account.

▪ **Define Categories**: This is where you tell the AI how to classify emails. Provide a **category name** and a **detailed description** to guide the model.

• **Category 1: Sponsorship**:

◦ Description: "Emails expressing interest in sponsoring, using words like 'sponsorship', 'deal', 'collaboration', 'paid partnership', etc.". (You can use ChatGPT to generate good descriptions).

• **Category 2: High Priority**:

◦ Description: "High priority emails often include words like 'urgent', 'time sensitive', 'deadline', or similar.". (You can add as many categories as you need).

▪ **Text to Classify**: Drag the **Text** output from the Gmail trigger node (under "Execute Previous Nodes" -> "Text") and drop it into this field. This feeds the email's body text to the classifier.

▪ Go "Back to Canvas".

◦ **Step 4: Add OpenAI Node for Sponsorship Email Response**.

▪ From the "Sponsorship" output of the "Text Classify" node, click the "+".

▪ Search for **"OpenAI"** and select **"OpenAI Model"**.

▪ **Operation**: Select **"Message a Model"**.

▪ **Model**: Select **"4o mini"**.

▪ Connect your OpenAI account.

▪ **AI Instructions (System Message)**: Provide clear instructions for the AI on how to respond.

• "You are an assistant of a YouTuber named Nafay 3D." (Customize with your or your client's name).

• "Your job is to respond to the email below. First, acknowledge the email, mention the sponsorship cost (e.g., $3000), offer flexibility for negotiation, and reconfirm interest in the sponsorship.". (This prompt was generated by ChatGPT).

• "The email to respond to is:" -> Drag the **Text** output from the "Text Classify" node into this field.

▪ **Role**: Set to **"System"** to define the model's behavior.

▪ **Output Content as JSON**: Turn **ON**. This formats the output in JSON, simplifying subsequent steps.

▪ **Fix for Response Format**: During testing, the AI might give a single block of text instead of distinct subject/body. To fix this, **add to your system prompt**: "I want you to respond in two parts: email subject and email body.".

▪ **Test Step**: Execute this step to see the AI-generated subject and body.

▪ Go "Back to Canvas".

◦ **Step 5: Add Gmail Node for Creating Sponsorship Draft**.

▪ Click the "+" button after the OpenAI response node for sponsorship.

▪ Search for **"Gmail"** and select **"Gmail Action"**.

▪ **Operation**: Select **"Create Draft"**. It's recommended to create a draft first instead of sending directly, to review and edit AI-generated responses before sending.

▪ **Subject**: Drag the **email_subject** output from the OpenAI node into this field.

▪ **Message**: Drag the **email_body** output from the OpenAI node into this field.

◦ **Step 6: Add OpenAI Node for High Priority Email Response**.

▪ From the "High Priority" output of the "Text Classify" node, click the "+".

▪ Add an **"OpenAI Model"** node with **"Message a Model"** operation.

▪ **Model**: Select **"4o mini"**.

▪ **AI Instructions (System Message)**:

• "You are an assistant. Your job is to respond to the following high priority email as good as you can.".

• "The email to respond to is:" -> Drag the **Text** output from the "Text Classify" node into this field.

▪ **Output Content as JSON**: Turn **ON**.

▪ **Fix for Response Format**: If needed, add "I want you to respond in two parts: email subject and email body" to the system prompt.

▪ **Test Step**: Execute and verify the generated response.

◦ **Step 7: Add Gmail Node for Creating High Priority Draft**.

▪ Click the "+" button after the OpenAI response node for high priority.

▪ Add a **"Gmail Action"** node.

▪ **Operation**: Select **"Create Draft"**.

▪ **Subject**: Drag the **email_subject** output from the previous OpenAI node.

▪ **Message**: Drag the **email_body** output from the previous OpenAI node.

2. **Final Testing and Delivery**:

◦ **Test Workflow**: Send a test sponsorship email and a test high-priority email to yourself. Run "Test Workflow" in n8n and then check your Gmail "Drafts" folder for the generated responses.

◦ **Activate**: Once satisfied, **activate** the workflow. It will now automatically process incoming emails in your Gmail account.

V. Project 3: RAG-based Data Analyst AI Agent

This project involves creating a chat model similar to ChatGPT but with the ability to answer questions directly from a custom PDF file (e.g., a financial report). This uses the **Retrieval Augmented Generation (RAG)** technique, a crucial skill in AI.

1. **Understanding RAG and Vector Databases**:

◦ **RAG (Retrieval Augmented Generation)**: An important AI technique.

◦ **Why Vector Databases?** AI models cannot directly read and understand normal PDF files. They need the information to be in a format they understand, which is a **vector database**.

◦ **Concept**: We convert the PDF file into **AI's language (vectors)** so the AI can read and comprehend it.

2. **Part 1: Uploading Data to the Vector Database (PDF to Vectors)**

◦ **Step 1: Add Trigger Node - Manually Trigger**.

▪ This part of the workflow will be run manually to ingest data.

◦ **Step 2: Prepare Your PDF File**:

▪ Find a PDF file (e.g., Apple's financial report).

▪ **Upload the PDF file to OneDrive** (or another cloud storage service).

◦ **Step 3: Pinecone Account Setup**:

▪ Go to **pinecone.io**.

▪ Click "Get Started" and **create a new account**. The free tier offers sufficient limits for this project.

▪ **Create an Index**:

• Navigate to **Databases -> Index** and click **"Create Index"**.

• **Name**: Give your index a descriptive name (e.g., ai-agents-course).

• **Configuration**: For the model, select **"text-embedding-3-small"**. This is an OpenAI embedding model.

• Leave other settings as default and click "Create Index". Wait for the index to build.

▪ **Obtain Pinecone API Key**:

• Go to **API Keys** on the left sidebar.

• If no key exists, click "Create New API Key", give it a name, and click "Create Key". **Copy the API key immediately**.

- **Step 4: Add OneDrive Node (Download File)**.
  - Click the "+" button after the "Manually Trigger" node.
  - Search for **"OneDrive"** and select **"OneDrive Action"**.
  - **Connect OneDrive Account**: Link your Microsoft OneDrive account.
  - **Resource**: Select **"File"**.
  - **Operation**: Select **"Download"**.
  - **File ID**:
    - Open your PDF file in OneDrive.
    - From the URL, copy the ID string that comes after ID= and before the next & symbol.
    - Paste this **File ID** into n8n.
  - **Test Step**: Execute this step. Verify that the file is downloaded. **Important**: Note that the file is in **Binary format**.
  - Go "Back to Canvas".
- **Step 5: Add Pinecone Vector Store Node (Add Documents)**.
  - Click the "+" button after the OneDrive node.
  - Search for **"Pinecone"** and select **"Pinecone Vector Store"**.
  - **Action**: Select **"Add Document to the Vector Store"**.
  - **Connect Pinecone Account**: Link your Pinecone account using the API key you copied.
  - **Operation**: Select **"Insert Document"**.
  - **Index Name**: Select the index you created (e.g., ai-agents-course).
  - **Add Option -> Pinecone Namespace**: Give it a name (e.g., apple-data). This organizes your data within the index.
  - **Embedding Model**: Select **"text-embedding-3-small"** (must match the model chosen during index creation in Pinecone).
  - Leave other settings as default.
  - **Document**: Select **"Default Data Loader"**. This tells n8n to load data from the previous OneDrive step.
  - **Type of Data**: Change this from JSON to **"Binary"** to match the output format from OneDrive.
  - Go "Back to Canvas".
- **Step 6: Add Recursive Character Text Splitter Node**.

- Click the "+" button after the "Default Data Loader" (which is connected to the Pinecone node).
- Select **"Recursive Character Text Splitter"**.
- **Purpose**: Large PDF files won't fit into a single AI prompt. This node **splits the document into smaller, manageable chunks** (e.g., pages) to make it easier for the AI model to process.
- Leave chunk size as default.
- Go "Back to Canvas".

- **Step 7: Execute Data Ingestion**:
  - Click **"Test Workflow"** on the "Manually Trigger" node.
  - Verify that the file is successfully stored in Pinecone by checking the "Name Spaces" section in your Pinecone dashboard. You should see your apple-data namespace appear.
  - At this point, your PDF data has been converted into **AI's vector language** and is ready to be used by an AI agent.

3. **Part 2: Retrieving Data with the AI Agent**
   - **Step 1: Create a New Workflow** and **Save** it.
   - **Step 2: Add Trigger Node - On Chat Message**.
     - This will be the primary trigger for interacting with your RAG agent.
   - **Step 3: Add AI Agent Node**.
     - Click the "+" button after the "On Chat" node.
     - Go to **"AI Advanced"** and select **"AI Agent"**.
     - Go "Back to Canvas".
   - **Step 4: Configure Chat Model (OpenAI)**.
     - Click the "+" next to "Chat Model".
     - Select **"OpenAI Chat Model"** and connect your account.
     - Go "Back to Canvas".
   - **Step 5: Configure Memory (Window Buffer Memory)**.
     - Click the "+" next to "Memory".
     - Add **"Window Buffer Memory"**.
     - Go "Back to Canvas".
   - **Step 6: Add Tool - Pinecone Vector Store (Retrieve Documents)**.
     - Click the "+" next to "Tool".
     - Search for **"Pinecone"** and select **"Pinecone Vector Store"**.
     - **Connect Pinecone Account**: Ensure your Pinecone account is correctly linked.

▪ **Operation Mode**: Select **"Retrieve Documents"**. (Previously, we used "Insert Document" for ingestion).

▪ **Description (Crucial!)**: This tells the AI model how and when to use this tool. Write a clear description: **"Use this tool to get the data to answer the user's question."** This ensures the AI retrieves answers from your PDF data rather than generating them from its general knowledge.

▪ **Pinecone Index**: Select your created index (e.g., ai-agents-course).

▪ **Add Options -> Pinecone Name Space**: Select the namespace you used earlier (e.g., apple-data).

▪ **Embedding Model**: Select **"text-embedding-3-small"**.

▪ Go "Back to Canvas".

◦ **Step 7: Test the RAG Agent**:

▪ **Activate** the workflow.

▪ Open the chat interface.

▪ **Ask a question directly from your PDF file**: For example, "What were the total net sales of Apple in September 30, 2023?" The agent should retrieve the precise information from the PDF and provide the correct answer (e.g., $89,000).

▪ **Ask for data analysis**: For example, "Analyze all the data and tell me pros and cons of investing in Apple." The agent will use the PDF data to perform an analysis, outlining sales, revenue sources, potential declines, valuation, and dependencies, similar to what a financial analyst would provide.

▪ **Potential**: This demonstrates the power of AI to automate complex data analysis, turning complex tasks into simple prompts.

VI. Conclusion and Future Learning

• AI is going to **change every industry significantly**. Adopting AI tools into your workflow is essential.

• n8n offers **over 300-400 useful tools and options** to explore beyond the projects covered in this course.

• The instructor encourages users to **continue their AI learning journey**.

• If there's demand for more advanced courses (e.g., voice agents, more complex agents), the instructor will create them. Stay updated by **subscribing to the channel, liking the video, and enabling notifications**.

Here is a summary document of the provided YouTube transcript, highlighting key concepts, tools, and projects covered in the course:

AI Agents Without Coding Full Course (n8n + OpenAI) - Summary Document

This course is designed for **beginners** with no prior coding or AI agents experience who want to learn how to build **realistic AI agents** using no-code platforms. The course is **completely free** on YouTube and aims to teach practical skills for building actual projects, potentially for freelancing.

1. Understanding AI Agents

• An **AI agent is a system that perceives its environment, processes information, and takes action to achieve a goal**, often using machine learning or rule-based logic.

• Examples of AI agents include **ChatGPT**, which takes input and provides output to achieve a user's goal (e.g., writing an essay), and **Instagram's recommendation algorithm**, which uses user feedback to suggest more content.

• AI agents are not new; they have been part of our lives for a long time and are becoming easier to create.

2. Introduction to n8n

• **n8n is a workflow automation tool** that enables users to automate tasks and integrate various APIs **without extensive coding**.

• It allows for the creation of "workflows," which are sequences of tasks automated by AI.

• **Key Benefits of n8n:**

  ◦ **No coding required** in 99% of cases, even for advanced applications.

  ◦ **Flexible workflow creation** with diverse options for different tasks.

  ◦ **Seamless integration with various AI APIs** like OpenAI, DeepSeek, and ClaudeAI.

• **Getting Started with n8n:**

  ◦ Users need to create an account on the n8n website.

  ◦ While n8n is a paid platform, a **free 14-day trial** is available without requiring a credit card, offering full feature access.

  ◦ The n8n user interface (UI) displays created workflows, projects, and settings.

  ◦ **Templates** are available as pre-made AI agents, but it's recommended to learn to build agents from scratch first to understand the logic.

  ◦ **Workflows are built using "nodes,"** which represent the steps an AI agent performs.

  ◦ **Triggers** define how an AI agent starts (e.g., manually, on a chat message, on a schedule, on an app event).

3. Core Components of AI Agents in n8n

• **Chat Model:** The Large Language Model (LLM) the AI agent uses, such as OpenAI's GPT-4o, GPT-4o mini, DeepSeek, or ClaudeAI. Options vary in cost and accuracy.

  ◦ **OpenAI API Integration:** Requires creating an account on platform.openai.com, adding credit, and generating an API key. The course recommends **GPT-4o mini** for its cost-effectiveness and good results.

• **Memory:** Essential for AI agents to remember previous interactions. The course uses **Window Buffer Memory**, which stores a set number of past messages (e.g., five) in the AI's memory. A higher context window length increases cost.

• **Tools:** Actions an AI agent can perform based on its input. Tools are integrated with various apps and services.

4. Real-Life Projects Covered

The course includes building three practical, realistic projects:

1. **Inventory Management System (with Google Sheets)**

◦ **Functionality:**

▪ Interacts with a **grocery database in Google Sheets** containing item names and quantities.

▪ Allows users to **query stock levels** (e.g., "How many bananas do we have?").

▪ Enables **updating inventory** (adding or removing items) via chat commands, automatically updating the Google Sheet.

◦ **Tools Used:** Google Sheets (for Get Rows and Append or Update Row operations), OpenAI Chat Model, Window Buffer Memory.

◦ **Smart Decision-Making:** The AI agent can decide whether to search the inventory or update it based on the user's message.

2. **Email Responder System (with Gmail and OpenAI)**

◦ **Functionality:**

▪ **Automatically triggers** upon receiving a new email in Gmail.

▪ **Reads and classifies incoming emails** into predefined categories (e.g., "Sponsorship," "High Priority") using OpenAI's model.

▪ **Generates a tailored response** based on the email's content and category.

▪ Can either **directly send the response or save it as a draft** for review, with drafting recommended due to potential AI inaccuracies.

◦ **Tools Used:** Gmail (trigger on "Message Received"), OpenAI Chat Model (for text classification and response generation), Text Classification node.

◦ **Customizable Categories:** Users can define any number of email categories and provide descriptions for how the AI should classify them.

3. **RAG-based Data Analyst AI Agent (with PDFs, OneDrive, Pinecone, and OpenAI)**

◦ **Functionality:**

▪ Allows interaction with **PDF documents** (e.g., financial reports) as if chatting with ChatGPT.

▪ Can **extract specific information** from the PDF (e.g., Apple's operating income).

▪ Capable of **performing data analysis** and answering complex questions based on the PDF content (e.g., "Pros and cons of investing in Apple").

◦ **Key Technique: RAG (Retrieval Augmented Generation)** – an important technique for enhancing AI responses with external data.

◦ **Two Main Parts:**

▪ **Storing Data in a Vector Database:** Converting PDF files into a format (vector database) that AI models can understand, as AI cannot directly read normal PDFs.

• **Tools Used:** OneDrive (to store PDF), Pinecone (to create an index and store vector data), Recursive Character Text Splitter (to break large files into smaller chunks for easier processing by the AI).

• **Pinecone Setup:** Create an account, an index (e.g., "AI-Agents-Course"), and generate an API key.

▪ **Retrieving Data using an AI Agent:**

• **Tools Used:** OpenAI Chat Model, Window Buffer Memory, Pinecone Vector Store (configured for Retrieve Document operation).

• The AI agent is instructed to use the Pinecone tool to retrieve data to answer user questions, ensuring responses are based on the provided document.

5. Deploying and Delivering AI Agents

• Once an agent is built and saved, it needs to be **activated** to function.

• For chat-based agents, you can make the chat **publicly available** via a URL.

• The initial chat message and appearance can be customized.

• Agents can be **embedded into other websites**, which typically involves providing instructions to a web developer.

6. Conclusion and Future Potential

• The course emphasizes the **immense potential of AI** to transform industries and workflows.

• Users are encouraged to **continue learning**, explore n8n's wide range of tools (over 300 options), and consider building more advanced AI agents, including voice agents.

• Adopting AI tools is crucial for staying relevant in the evolving technological landscape.