



appian



Appian AI Challenge

Project-1

Classification and processing of unstructured financial documents

Access links

Pradipta Sundar Sahoo
Indian Institute of Technology (IIT) Roorkee
pradipta_ss@ch.iitr.ac.in

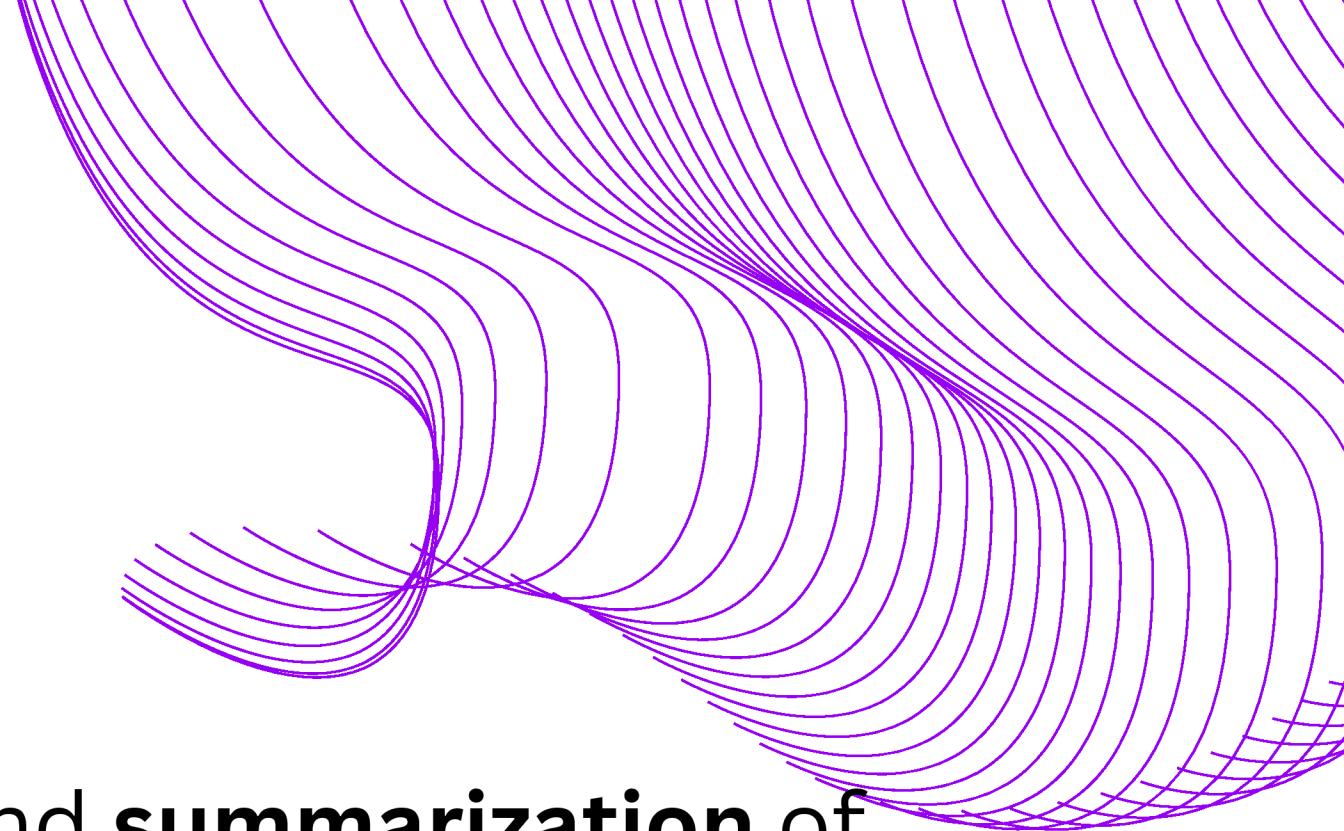
Agenda

- About the project
- Features
- Proposed Solution
- Results
- Access links
- Future Advancements
- Setup Instructions
- Usage
- DB Architecture

Tip: Use links to go to a different page inside your presentation.

How: Highlight text, click on the link symbol on the toolbar, and select the page in your presentation you want to connect.

About



- Goal: To automate the **classification, association, and summarization** of **unstructured financial documents**, enabling faster processing and reducing manual effort while maintaining accuracy and scalability.
- Key Objectives:
 - Document Categorization
 - Summarization
 - Scalability and Efficiency, Ease of Use, Accuracy and Reliability

[BACK TO AGENDA PAGE](#)

Key Features



Caching

Speeds up access to processed data using Redis for frequently accessed information.



OCR Integrated Tool

Extracts text efficiently from PDFs and images using robust OCR tools.



Hierarchical Classification

Organizes documents by individual, type, and subcategories



Vector Search

Handles large volumes of FAQs and multiple user queries.

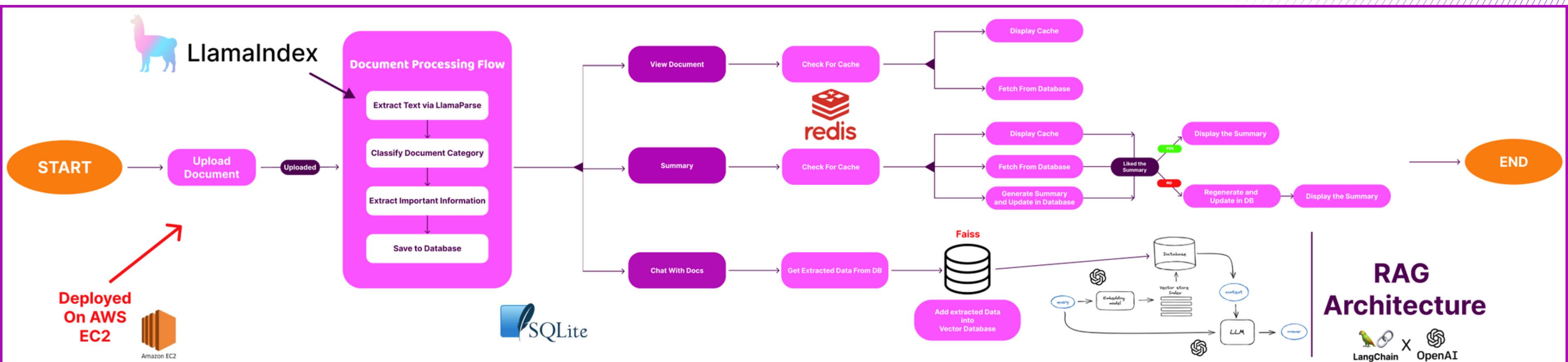


Database Management

Stores structured data securely with SQLite for auditability and redundancy.

[BACK TO AGENDA PAGE](#)

Proposed Solution



[BACK TO AGENDA PAGE](#)

Techniques used

- **Recursive Text Splitting:** Uses RecursiveCharacterTextSplitter for breaking down documents into manageable chunks while maintaining context
- **Retrieval Augmented Generation (RAG):** Implements a hybrid approach combining document retrieval with LLM generation for accurate document-based chat
- **Vector Similarity Search:** Employs FAISS for efficient document similarity matching and retrieval
- **Caching Strategy:** Uses a multi-level caching system combining Redis for distributed caching and local LRU cache for frequently accessed data
- **State Management:** Implements custom state handling for chat sessions and document processing workflows.

Technology used

- **LlamaParse:** For robust PDF and document text extraction
- **OpenAI GPT-4:** Powers the document classification, summarization, and chat functionality
- **Streamlit:** Frontend framework for the web interface
- **Redis:** Distributed caching system for document data and chat history
- **SQLite:** Local database for document storage and metadata
- **FAISS:** Vector database for efficient similarity search
- **LangChain:** Framework for building the RAG pipeline
- **LangGraph:** For managing conversational workflows and state



Streamlit

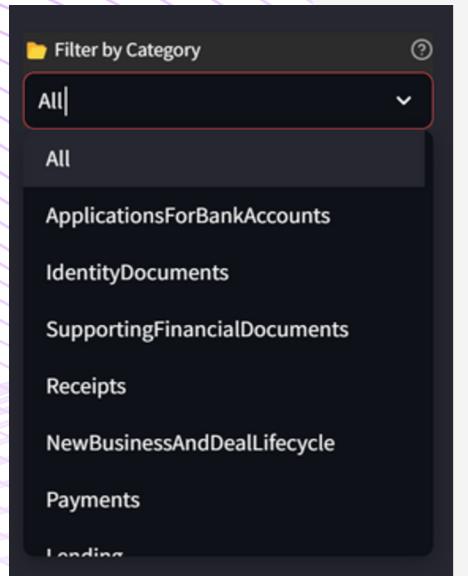


[BACK TO AGENDA PAGE](#)

Results

Output

You can select the category of documents you have uploaded earlier



1. Upload Document

The screenshot shows the 'Document Viewer' interface. In the center, there's a banner for 'SMART <financial doc> <classifier and processor>' with the Appian logo. Below it, the 'Upload New Document' section shows a file named 'Invoice_udemy.pdf' being uploaded. A green success message says 'Document Processed Successfully!' with 'Main Category: Payments' and 'Subcategory: Invoices'. On the left, there's a mobile device icon showing a document viewer interface.

2. View Document

The screenshot shows the 'Document Viewer' interface. It displays the same 'SMART <financial doc> <classifier and processor>' banner. In the 'View Document' section, it shows the same 'Invoice_udemy.pdf' file with its details: 'Upload Date: 2024-12-23 05:28:15.113583', 'Main Category: Payments', and 'Subcategory: Invoices'. Below this, there's a 'Retrieved document from cache' message. To the right, there's a 'Summarize' button and a 'Chat' button. At the bottom, there's a section titled 'Important Information' with a list of details including 'Invoice #: IN2024351007', 'Invoice date: 17/12/2024', 'Company information: Udemy India LLP', 'Company address: 10th Floor, ResCowork 07, Tower B, Urile Cybei Paik Seclo 39, Gurgaon, Haryana, India, 122003', 'GSTIN: OBAAFFU9783MM1ZE', 'FAN no: AAFFU9763M', and 'Personal identifier: Pradipa Sundar Sahoo'.

3. Summarise

The screenshot shows the 'Document Summary' interface. It starts with a summary of the document: 'Invoice_udemy.pdf', 'Upload Date: 2024-12-23 05:28:15.113583', 'Main Category: Payments', and 'Subcategory: Invoices'. Below this, there are three buttons: 'View Document', 'Summarize', and 'Chat'. The 'Summarize' button is highlighted. The main content area is titled 'Document Summary' and contains a 'Summary of Invoice Document' section. It details the 'Document Type' (Tax Invoice (Original for Recipient)), 'Invoice Details' (Invoice Number: IN2024351007, Invoice Date: 17/12/2024), 'Supplier Information' (Name: Udemy India LLP, Address: 10th Floor, ResCowork 07, Tower B, Urile Cybei Paik Seclo 39, Gurgaon, Haryana, India, 122003, GSTIN: OBAAFFU9783MM1ZE, FAN Number: AAFFU9763M), and 'Additional Notes' (This is a system-generated invoice and does not require a physical or digital signature). At the bottom, there are 'Copy Summary' and 'Regenerate' buttons, with a red arrow pointing to the 'Regenerate' button.

4. Chat With the Doc

The screenshot shows the 'Chat with Document' interface. It displays the same document details: 'Invoice_udemy.pdf', 'Upload Date: 2024-12-23 05:28:15.113583', 'Main Category: Payments', and 'Subcategory: Invoices'. Below this, there are three buttons: 'View Document', 'Summarize', and 'Chat'. The 'Chat' button is highlighted. The main content area is titled 'Chat with Document' and contains a placeholder 'Ask questions about the document content'. A question is asked: 'What was the course and who was the buyer?'. The AI response is: 'The course was "Complete Guide to Elasticsearch" and the buyer was Pradipa Sundar Sahoo.' At the bottom, there is a 'Clear Chat History' button.

You can regenerate the summary

Tap to view the document and important info

Bonus Video:

[Bonus Video Link](#)

I was working on a finance project myself, which is not published yet anywhere (till now).

Have A Look!!!
120 page Bank statement, with lots of numbers

Live Access Link:

[Appian AI Fin Document](#)

Demo Video Link:

[Demo Video Link](#)

Access Links

Github Link:

[Github Repo Link](#)

[BACK TO AGENDA PAGE](#)

Evaluation?

The **most important thing**.

Have updated evaluation.py

But need to create a test dataset manually.

Which I will do on moving to next round.

LlamaParse		
<small>Apple Inc. CONDENSED CONSOLIDATED STATEMENTS OF OPERATIONS (Unaudited) (In millions, except number of shares, which are reflected in thousands, and per-share amounts)</small>		
	Three Months Ended September 30, 2023	Two Months Ended September 30, 2022
Net sales:		
Products	\$ 67,044	\$ 70,946
Services	21,914	19,568
Total net sales ⁽¹⁾	\$ 89,458	\$ 80,514
Cost of sales:		
Products	42,586	46,387
Services	6,485	6,464
Total cost of sales	\$ 49,071	\$ 52,851
Gross margin		
	40,427	38,068
Operating expenses:		
Research and development	2,607	2,781
Selling, general and administrative	8,193	8,243
Total operating expenses	13,658	13,201
Operating income:		
Operating income	26,869	24,894
Other income/(expense), net	29	(237)
Income before provision for income taxes	26,898	24,657
Provision for income taxes	4,542	3,898
Net income		
	\$ 22,356	\$ 20,759
Earnings per share:		
Basic	\$ 1.47	\$ 1.29
Diluted	\$ 1.46	\$ 1.29
Shares in computing earnings per share:		
Basic	15,989,434	16,030,382
Diluted	15,872,400	16,016,465
Net sales by reportable segment:		
Americas	\$ 40,115	\$ 39,808
Europe	22,463	22,795
Greater China	16,262	17,512
Japan	8,509	8,700
Rest of Asia Pacific	6,331	6,373
Total net sales		
	\$ 89,458	\$ 80,514
Net sales by category:		
iPhone	\$ 43,805	\$ 42,626
Mac	7,014	7,168
iPad	6,443	7,174
Wearables, Home and Accessories	8,342	8,499
Services	21,914	19,568
Total net sales		
	\$ 89,458	\$ 80,514
<small>⁽¹⁾ Net sales by category: iPhone, Mac, iPad, Wearables, Home and Accessories, Services.</small>		

PyPDF

PyPDF	
<small>Apple Inc. CONDENSED CONSOLIDATED STATEMENTS OF OPERATIONS (Unaudited) (In millions, except number of shares, which are reflected in thousands, and per-share amounts)</small>	
	Three Months Ended September 30, 2023
Net sales:	
Products	\$ 67,044
Services	21,914
Total net sales ⁽¹⁾	\$ 89,458
Cost of sales:	
Products	42,586
Services	6,485
Total cost of sales	\$ 49,071
Gross margin	
	40,427
Operating expenses:	
Research and development	2,607
Selling, general and administrative	8,193
Total operating expenses	13,658
Operating income:	
Operating income	26,869
Other income/(expense), net	29
Income before provision for income taxes	26,898
Provision for income taxes	4,542
Net income	
	\$ 22,356
Earnings per share:	
Basic	\$ 1.47
Diluted	\$ 1.46
Shares in computing earnings per share:	
Basic	15,989,434
Diluted	15,872,400
Net sales by reportable segment:	
Americas	\$ 40,115
Europe	22,463
Greater China	16,262
Japan	8,509
Rest of Asia Pacific	6,331
Total net sales	
	\$ 89,458
Net sales by category:	
iPhone	\$ 43,805
Mac	7,014
iPad	6,443
Wearables, Home and Accessories	8,342
Services	21,914
Total net sales	
	\$ 89,458
<small>⁽¹⁾ Net sales by category: iPhone, Mac, iPad, Wearables, Home and Accessories, Services.</small>	

LlamaParse Wins !!!



GPT-4o , undoubtedly one of the best

Metrics	Baseline (PyPDF + Naive RAG)	LlamaParse + Recursive Retrieval
mean_correctness_score	3.874	4.270
mean_relevancy_score	0.844	0.926
mean_faithfulness_score	0.667	0.915

Results over the [Uber 10K Dataset](#). For more information on our evaluation metrics check out our [evaluation page](#) here.

Future Advancements

Multimodal input and output:

have already been implemented. ([in repo](#)). But do we actually need it?

Multilingual:

knows how to implement. Could be useful !!!.



Faster Inference :

could use small llms.

Scalability:

have already implemented a [FastAPI](#) code. It could be used for production. ([in repo](#)).

Relevance:

Reranked all results according to relevance. May a more robust way be done.

More robust:

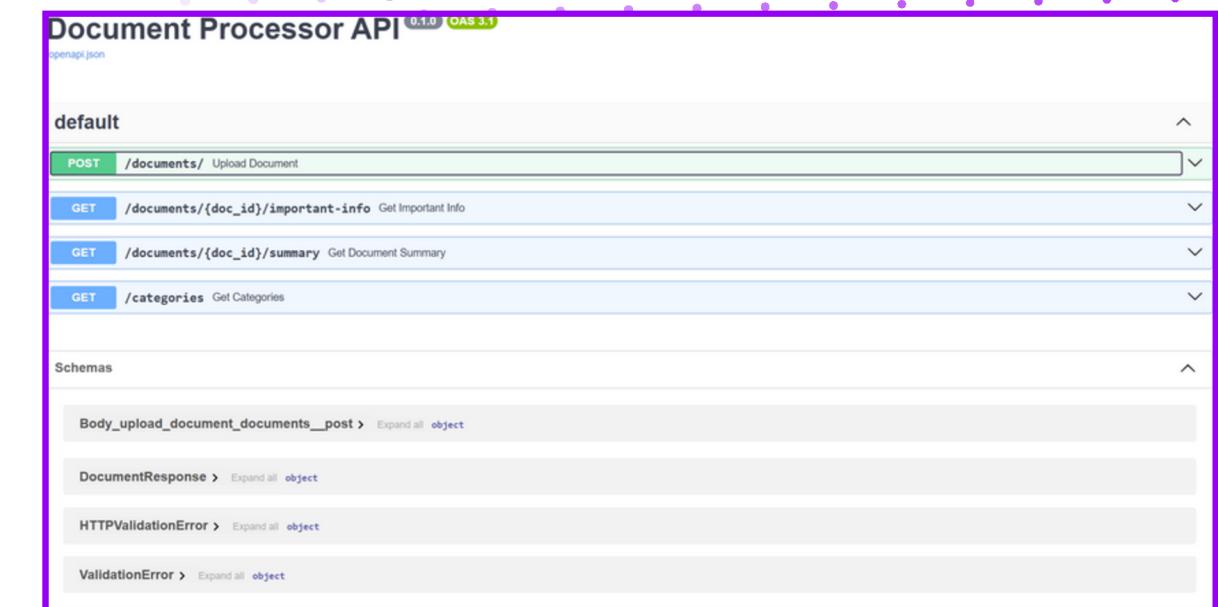
I have already some extra info about different categories that APPIAN could track in future.

But more info can be fed.

Cost-effective:

Instead of EC2, can be deployed in serverless instances.

More intermediate stuffs can be removed to lower the tokens count.



WHAT ELSE I HAVE TRIED?

- Tried with pinecone (fetching time was more)
- Tried Agentic Frameworks (Takes a lot of tokens and latency is high, but no such improvements)

Setup Instructions

[Github Repo Link](#)

[BACK TO AGENDA PAGE](#)

Prerequisites

- Python 3.8+
- Redis Server
- OpenAI API Key
- LlamaParse API Key

Installation

1. Clone the repository:

```
git clone <repository-url>
cd document-categorizer
```

2. Install required packages:

```
pip install -r requirements.txt
```

3. Set up environment variables:

```
OPENAI_API_KEY=your_openai_api_key
LLAMA_CLOUD_API_KEY=your_llama_parse_api_key
```

4. Start Redis server:

```
redis-server
```

5. Run the application:

```
streamlit run app.py
```

Usage

[Github Repo Link](#)

Usage

1. Upload Document:

- Click on the file uploader
- Select a PDF document
- Click "Process Document"

2. View Documents:

- Browse through uploaded documents in the Document History section
- Filter documents by category
- View document details and content

3. Document Analysis:

- View automatic categorization results
- See extracted important information
- Generate and read document summaries

4. Interactive Features:

- Chat with documents using AI
- Download processed documents
- View document summaries

• Database Architecture:

```
-- Main Document Table
CREATE TABLE documents_new (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    filename TEXT NOT NULL,
    category TEXT NOT NULL,
    subcategory TEXT NOT NULL,
    important_info TEXT,
    pdf_data BLOB,
    upload_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    all_extracted TEXT,
    summary TEXT
);

-- Document Versions Table
CREATE TABLE document_versions (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    doc_id INTEGER,
    version_number INTEGER,
    changes TEXT,
    modified_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (doc_id) REFERENCES documents_new(id)
);

-- Chat History Table
CREATE TABLE chat_history (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    doc_id INTEGER,
    user_message TEXT,
    ai_response TEXT,
    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (doc_id) REFERENCES documents_new(id)
);
```

DB Architecture

[Github Repo Link](#)

BACK TO AGENDA PAGE

[BACK TO AGENDA PAGE](#)

Thank You

About Me

- Avid Learner with strong problem solving skills.
- Keep tracks of recent GenAI Hacks
- Worked with multiple startups (healthtech, legaltech, fintech) build their MVP.
- Collaborated with cross-domain teams, showcasing good leadership and team management skills.
- Currently, focussing on AI Agents Architecture.



Pradipta Sundar Sahoo
UG (III Year I Semester)
B.Tech. (Chemical Engineering)
Contact No: 7327873628
Email: pradipta_ss@ch.iitr.ac.in
Registration No: 22118054/2025



Education

Year	Degree/Examination	Institution/Board	CGPA/Percentage
2024	B.Tech. 2nd Year	Indian Institute of Technology, Roorkee	8.271
2022	Intermediate (Class XII)	Shiv Jyoti Senior Secondary School , Kota	94.00 %
2020	Matriculate (Class X)	Vivekananda Shiksha Kendra , Bhubaneshwar	93.20 %

Experience

Co-founder | Lawcrats

August 2024 - October 2024

- Co-founded a legal tech AI startup, led a 12-member team, and secured support from Google, Microsoft, AWS, MongoDB; approved for NVIDIA Inception Program and Wadhwan Foundation cohort'24.
- Engaged with lawyers and advocates through meetings and discussions to align technology solutions with legal industry needs. I had to learn and implement a new technology.

Internships

AI Engineer intern | Anguliaym

August 2024 - October 2024

- I have developed end-to-end AI healthcare RAG applications for structured outputs, further incorporating data extraction, AI app development, and model deployment.
- Created conversation voice bot (s2s) using NixTTS and local llms. Reducing traditional latency from 32 s to 2s.

AI Engineer Intern | Tecosys (Acquired by Medynex)

June 2024 - August 2024

- Engineered and optimized AI RAG systems and tools, integrating Gen AI with Python to deliver scalable, high-performance applications, reducing costs and enhancing efficiency.
- Demonstrated creativity, problem-solving, and teamwork, consistently delivering innovative solutions under pressure with precision and attention to detail.

Projects

Praml An Automl | Indian Institute of Technology Roorkee

April 2024 - May 2024

- Created Praml, a Python Automated Machine Learning Library, to streamline building ML models by automating data preprocessing, model selection, hyperparameter tuning, and evaluation.
- Integrated diverse models for classification and regression with custom-made Tree-Parzen Estimator and hyperopt optimization.

Extreme low light denoiser | Indian Institute of Technology Roorkee

May 2024 - June 2024

- Developed an extreme low-light denoiser using multiple architectures including FFDNet, SwinIR, ResNet and UNet.
- ResNet emerged as the best-performing architecture with a PSNR of 25 after 40 epochs, significantly improving image quality in low-light conditions.

Monte Carlo Simulation Using Python | Indian Institute of Technology Roorkee

October 2023 - November 2023

- Implemented a Monte Carlo simulation in Python utilizing the Sharpe ratio to analyze and optimize investment strategies.
- Generated graphs to visualize simulated portfolio returns and risk-adjusted performance, providing insights into optimal asset allocation and risk management strategies.

Object detection using Opencv and Yolov8 | Indian Institute of Technology Roorkee

February 2024

- Developed an object detection system to identify and classify objects in images and video streams.
- Utilized OpenCV for image processing and integrated YOLOv8 for real-time detection.
- Enhanced accuracy and efficiency in surveillance, autonomous vehicles, and robotics applications.

Stock Sentiment Analysis Using Machine Learning | Indian Institute of Technology Roorkee

May 2024 - June 2024

- Developed a system to predict stock price movements by analyzing sentiment from financial news headlines using machine learning.
- Scrapped financial news data using Beautiful Soup and visualized trading signals on stock price charts to enhance decision-making.

Skills

Computer languages

Python, C++, Js

Software Packages

PyTorch, TensorFlow, Linux, Git ,CI/CD, Docker, Kubernetes, PostgreSQL, MongoDB, OpenAI, Ollama, Langchain, Tableau, Airflow, REST APIs, GraphQL, Flask, FastAPI, CI/CD, AWS, Azure

Positions of Responsibility & Extra Curriculars

October 2024

Delegate | FICCI India AI

- Invited as a delegate to the FICCI India AI Conclave'24.
- Engaged in discussions with co-founders and heads from Google, AWS, and Panasonic on AI development initiatives.