AI Learning & Hackathons

As part of my extracurricular activities beyond my university coursework, I took courses from platforms like DeepLearning.AI, Coursera, Udemy, and Kaggle, and also participated in online hackathons on Kaggle to enhance my understanding and practical skills. I have earned certifications in Machine Learning, Deep Learning, Computer Vision, Video Processing, Natural Language Processing, Reinforcement Learning, and Generative AI. I've worked on various projects in these areas, including chatbots, large language models (LLMs), vision-language models (VLMs), Transformers, and video processing. To deepen my understanding, I also built machine learning (ML) and deep learning (DL) models from scratch, without using tools like Scikit-learn, PyTorch, or TensorFlow. These activities have helped me strengthen both my theoretical knowledge and practical skills. A few of my projects and certificates are listed below.

Kaggle Profile Overview

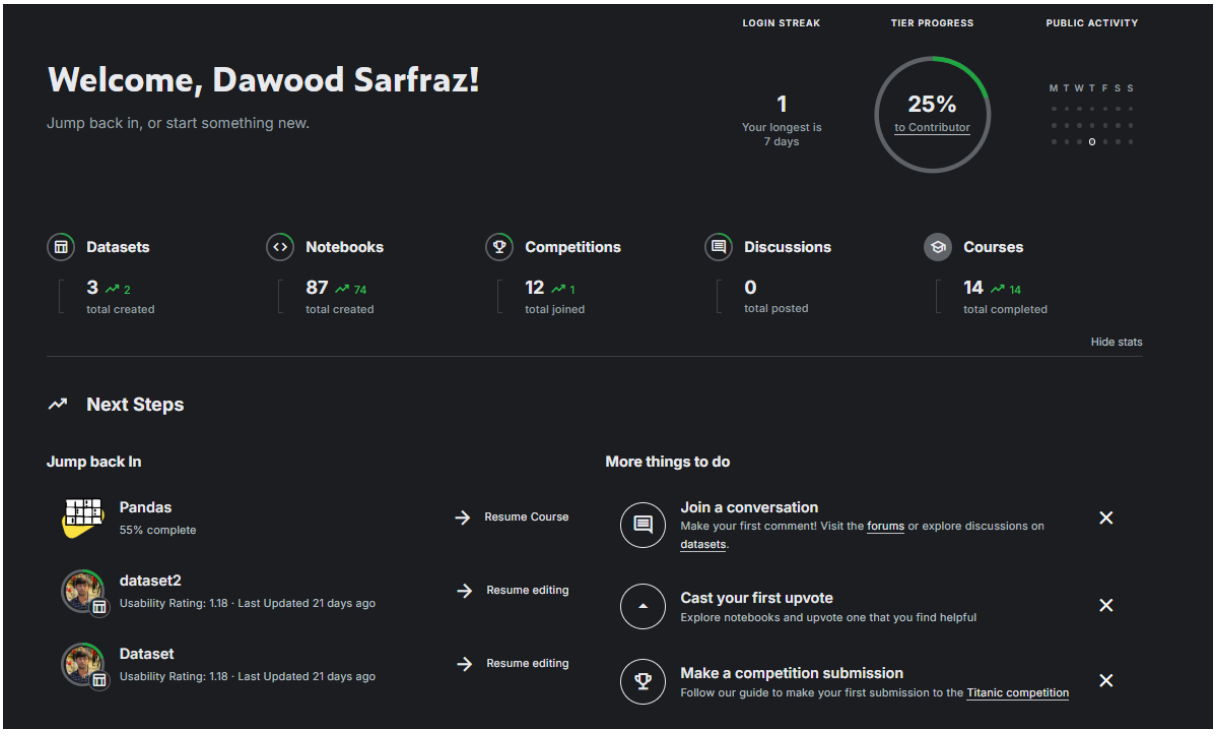

Figure 1: Kaggle Profile overview

Projects Overview

- **Smart Surveillance System**
  In this project, I have developed a real-time Traffic Analysis System that processes live RTSP video streams by converting them to WebRTC for efficient web-based analytics. I implemented advanced object tracking algorithms like SORT, DeepSORT, and ByteTrack to monitor movement and estimate direction. The system comprises modules for object detection, direction estimation, and vehicle counting, enabling the tracking of congestion. Additionally, I built algorithms for wait time estimation, queue detection, speed calculation, and traffic light violation detection to enhance traffic flow and ensure rule enforcement. The project also features line and zone intrusion detection to improve security and manage restricted areas.
  GitHub: https://github.com/Dawoodsarfraz/Smart-Surveillance-System

- **Deep Learning Approaches for Multi-Class Cancer Classification: A Comparative Study of CNN, ShuffleNet, and NasNet Models**
  Skin cancer, a prevalent form of cancer, impacts millions of individuals annually due to abnormal cell growth. This can lead to a dangerous spread to other parts of the body, which can be life-threatening. Early detection and staging are crucial because skin cancer has high mortality rates. We have done extensive studies to identify the best classifier suitable for dermoscopic image classification using different deep neural networks. Various deep neural network architectures, including CNN, NasNet, and ShuffleNet models, were evaluated using the HAM10000 dataset. The dataset was obtained from the Harvard Dataverse and comprises seven distinct types of lesions. To improve model performance, we used RandomOverSampler to address data imbalance. Results showed promising outcomes: CNN achieved 92% accuracy, NasNet improved further to 93%, and ShuffleNet reached 87%. These findings highlight the potential of advanced neural networks to enhance skin cancer diagnosis.
  GitHub: `https://github.com/Dawoodsarfraz/Deep-Learning-Approaches-for-Multi-Class-Cancer-Classification`

- **Gastrointestinal Disease Classification using Endoscopic Images**
  In this project, I worked on automatic disease detection using endoscopic images from the Kvasir dataset, which contains labeled gastrointestinal (GI) tract images, annotated by experienced medical professionals. For this task, I implemented and experimented with hybrid deep learning architectures, combining the strengths of EfficientNet, RegNet, Inception, ResNet152, and VGG19 with transformer-based models like ViT and Swin Transformer, to capture both local and global features effectively for enhanced performance in medical image classification. These hybrid models were designed to leverage the spatial precision of CNNs and the contextual reasoning power of transformers, providing a robust solution for automated GI disease detection.
  GitHub: `https://github.com/Dawoodsarfraz/Gastrointestinal-Disease-Classification-using-Endoscopic-Images`

- **RAG System with Llama 3.2**
  This project implements a Retrieval-Augmented Generation (RAG) system that integrates document retrieval and text generation to provide accurate, context-aware answers. It uses LangChain to orchestrate the retrieval and generation workflow, Ollama for efficient model deployment, and Fasiss CPU for energy-efficient inference. PyPDF extracts relevant text from PDFs for the retrieval phase, while MTEB ensures high-quality embeddings for improved accuracy. The system is optimized for tasks like research, customer support, and knowledge

management, offering dynamic and precise responses.
GitHub: https://github.com/Dawoodsarfraz/RAG-Llama-3.2

- **InsightFlow**
  In this project, I am building InsightFlow, an AI-driven system for real-time news retrieval, research, and content creation. It utilizes four core agents: News Retriever, News Scraper, News Summarizer, and News File Writer to efficiently gather, analyze, summarize, and write articles on any given topic. The project integrates tools like Ollama for context-aware summaries, Llama 3.2 for processing complex news data, CrewAI for task orchestration, and Serper for retrieving up-to-date articles. Together, these components streamline the news gathering and content generation process, ensuring timely and relevant information.
  GitHub: https://github.com/Dawoodsarfraz/InsightFlow

- **LLamaAssist**
  LLamaAssist is an AI-powered application built with Llama 3.2-3B and using Ollama. It simplifies content creation, offering customizable options for assignments, blogs, and AI-driven interactions. It leverages LangChain for advanced language model orchestration and Streamlit for an interactive user interface. Key features include content generation, file saving, and downloading, document interaction (QnA with PDFs), and image chatting with Llama 3.3 Vision. LLamaAssist offers a seamless and efficient tool for content creation and document management.
  GitHub: https://github.com/Dawoodsarfraz/LlamaAssist

- **Cyber Attacks Classification using Machine Learning**
  In the rapidly evolving landscape of cybersecurity, the ability to quickly and accurately classify cyber attacks is crucial. This project leverages machine learning algorithms to analyze and classify patterns in network traffic, helping to identify and mitigate potential threats. The project includes several machine learning models for cyber attack classification, like Decision Trees, Random Forests, Support Vector Machines (SVM), and Deep Neural Networks.
  GitHub: `https://github.com/Dawoodsarfraz/Cyber-Attacks-Classification-using-Machine-Learning`

- **Text, Image, Audio and Video Steganography**
  This is an encryption and decryption software developed using Python 3 on the Linux platform. This desktop application can encrypt and decrypt simple text files(e.g., .txt, .py), Image files(8, 16, 24 bit depth jpg, jpeg, bmp, png, etc), Audio files(8 and 16-bit Mono and Stereo PCM files), Video files(.avi, .mp4). This application uses the RSA algorithm to perform encryption and decryption of files, though for image and audio files, the algorithm is enhanced to make the encryption and decryption more efficient.
  GitHub: https://github.com/Dawoodsarfraz/Text-Image-Audio-and-Video-Steganography

- **RoboText Classifier**
  This project leverages RoBERTa and NLTK for text classification on a large 500k-row dataset, aiming to build a model that efficiently classifies text using advanced techniques. The model utilizes dynamic masking to improve generalization by randomly masking tokens during training, enhancing its ability to handle varied contexts. Sentence packing is employed to combine multiple sentences into a single input sequence, maximizing the 512-token input limit and enabling the model to process longer documents. Larger batch sizes are used to optimize training speed and stability, allowing for the efficient processing of multiple samples at once.

Finally, the model uses a byte-level BPE vocabulary to tokenize text at the byte level, ensuring robust handling of diverse text, including special characters and misspellings.
GitHub: https://github.com/Dawoodsarfraz/RoboText-Classifier

- **Named Entity Recognition (NER) system using Hugging Face models**
  I built a Named Entity Recognition (NER) system using Hugging Face models. The goal is to identify and classify entities such as people, organizations, locations, dates, and other relevant information from text. I am utilizing DistilBERT, a smaller and faster variant of BERT, for NER tasks. The project leverages Hugging Face's transformers library, where the model is initialized using the AutoTokenizer for tokenizing the input text. The pre-trained model, distilbert-base-cased, is fine-tuned for NER tasks to extract named entities from various sources of text efficiently.
  GitHub: `https://github.com/Dawoodsarfraz/Name-Entity-Recognization-using-HuggingFace`

- **Intellicrew**
  I developed Intellicrew, an AI-powered multi-agent system built using the CrewAI framework. Intellicrew is designed to streamline complex workflows by coordinating a team of intelligent agents, such as the Researcher and Reporting Analyst, each specialized in handling specific tasks. These agents collaborate autonomously to gather, process, and present information efficiently. The system is powered by Llama 3.2, hosted locally via Ollama, enabling fast, secure, and context-aware language generation. The entire backend is implemented in Python, with optional support for the OpenAI API for enhanced processing capabilities.
  GitHub: https://github.com/Dawoodsarfraz/Intellicrew

- **Electronic Products Recommendation System**
  In this project, I built a recommendation system using the Amazon Electronics dataset to enhance product suggestions based on user interactions and product metadata. The dataset contains over 7 million user-product interactions, including ratings and timestamps. I applied techniques like collaborative filtering, content-based filtering, and matrix factorization to model user preferences and item similarities. Additionally, I implemented hybrid methods to combine these approaches, improving both the accuracy and diversity of recommendations. The system aims to deliver personalized product suggestions, ultimately boosting user satisfaction and engagement in e-commerce platforms.
  GitHub: https://github.com/Dawoodsarfraz/electronic-products-recommendation-system

- **Stock Market Prediction using LSTM**
  In this project, we train an LSTM model using historical stock market data to predict future stock prices based on past trends. The dataset includes features like opening/closing prices and volume, which are preprocessed into training and testing sets. We used stock data from different companies, such as Microsoft, fetched using Yahoo Finance via the yfinance library. The model is built using TensorFlow, with optimized hyperparameters to improve performance and prevent overfitting.
  GitHub: https://github.com/Dawoodsarfraz/Stock-Market-Prediction-using-LSTM

Intro to AI Ethics

Figure 2: Intro to AI Ethics

Machine Learning Explainability

Figure 3: Machine Learning Explainability

Intro to Machine Learning

Figure 4: Intro to Machine Learning

Intermediate Machine Learning

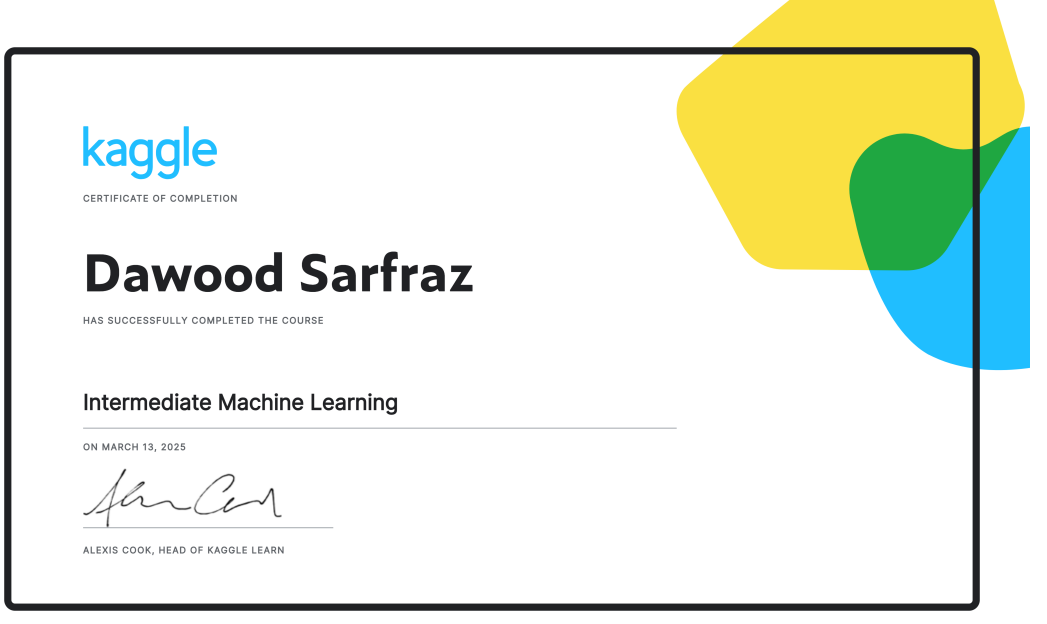

Figure 5: Intermediate Machine Learning

Intro to Deep Learning

Figure 6: Intro to Deep Learning

Computer Vision

Figure 7: Computer Vision

Time Series

Figure 8: Time Series

Intro to Game AI and Reinforcement Learning

Figure 9: Intro to Game AI and Reinforcement Learning

Data Cleaning

Figure 10: Data Cleaning

Feature Engineering

Figure 11: Feature Engineering

Data Visualization

Figure 12: Data Visualization

Geospatial Analysis

Figure 13: Geospatial Analysis

Introduction to Machine Learning

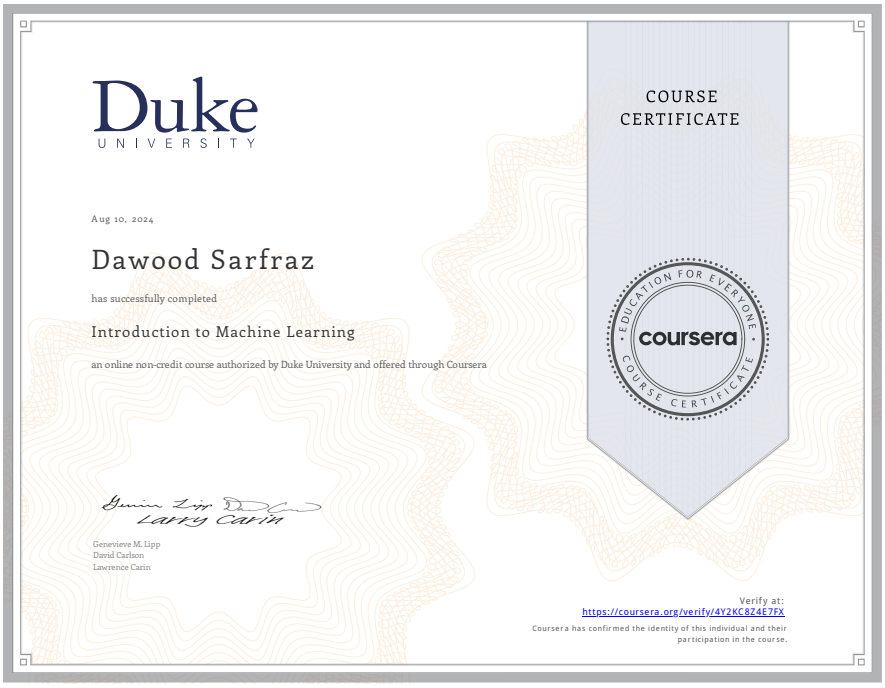

Figure 14: Introduction to Machine Learning

Supervised Machine Learning Regression and Classification

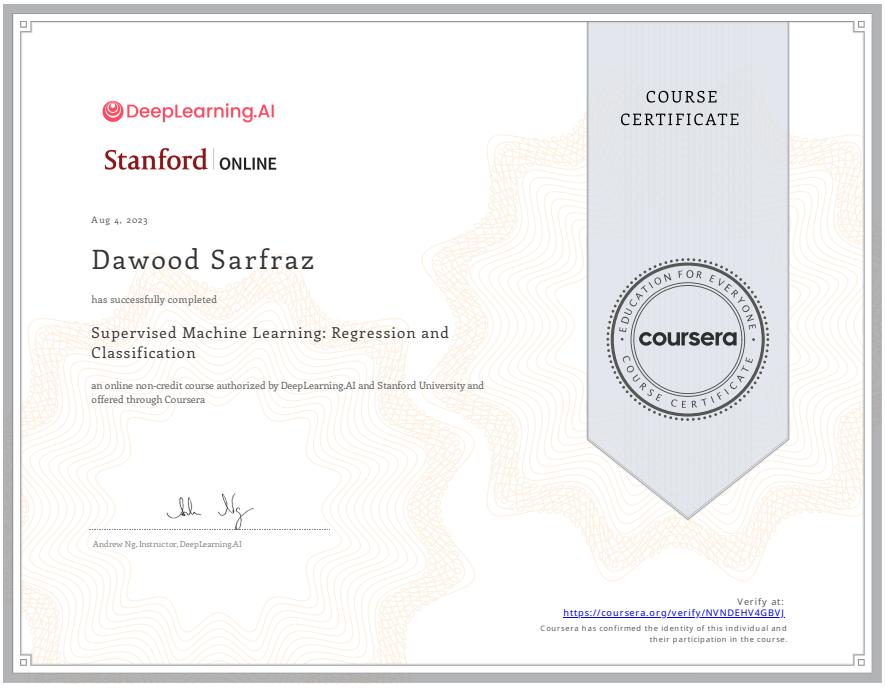

Figure 15: Supervised Machine Learning Regression and Classification

Natural Language Processing in TensorFlow

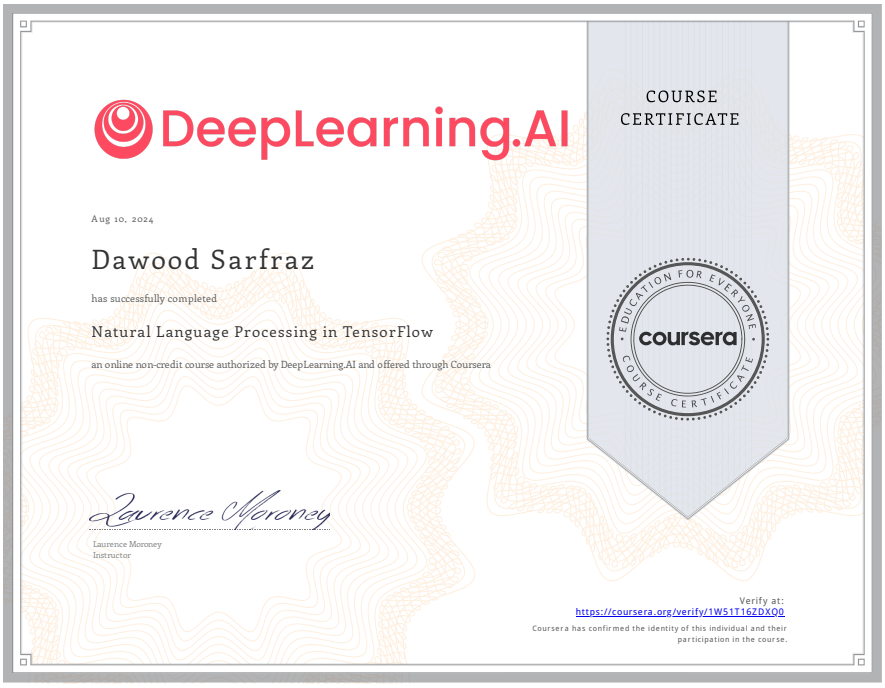

Figure 16: Natural Language Processing in TensorFlow

Sentiment Analysis, Beginner to Expert

Figure 17: Sentiment Analysis, Beginner to Expert

Project-Based Text Mining in Python


udemy

Certificate no: UC-89713b43-55dc-4e9e-924e-3b8367f5efba
Certificate url: ude.my/UC-89713b43-55dc-4e9e-924e-3b8367f5efba
Reference Number: 0004

CERTIFICATE OF COMPLETION

Project based Text Mining in Python

Instructors **Taimoor khan**

Dawood Sarfraz

Date **Oct. 26, 2023**
Length **10.5 total hours**

Figure 18: Project-Based Text Mining in Python

MATLAB Master Class: Go from Beginner to Expert


udemy

Certificate no: UC-8ef1a1a9-c535-43b1-bf89-8ff81d1c4df0
Certificate url: ude.my/UC-8ef1a1a9-c535-43b1-bf89-8ff81d1c4df0
Reference Number: 0004

CERTIFICATE OF COMPLETION

MATLAB Master Class: Go from Beginner to Expert in MATLAB

Instructors **Nouman Azam**

Dawood Sarfraz

Date **July 31, 2023**
Length **50.5 total hours**

Figure 19: MATLAB Master Class: Go from Beginner to Expert

How Transformer LLMs Work

Figure 20: How Transformer LLMs Work

Attention in Transformers Concepts and Code in PyTorch

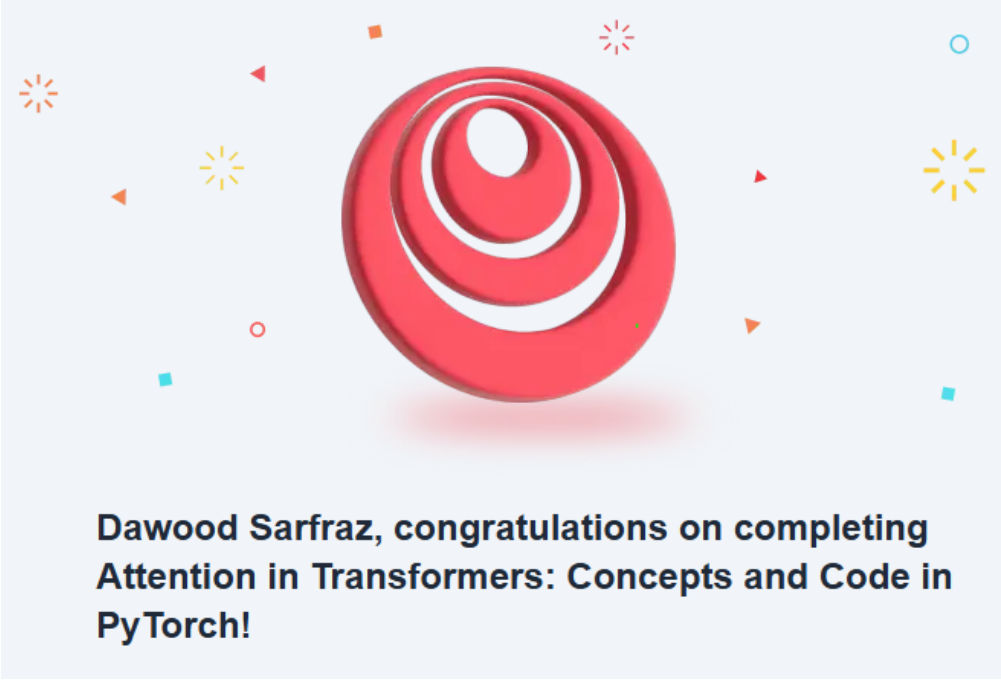

Figure 21: Attention in Transformers Concepts and Code in PyTorch

Getting Started with Mistral

Figure 22: Getting Started with Mistral

Quantization Fundamentals with Hugging Face

Figure 23: Quantization Fundamentals with Hugging Face

Quantization in Depth

Figure 24: Quantization in Depth

LangChain Chat with Your Data

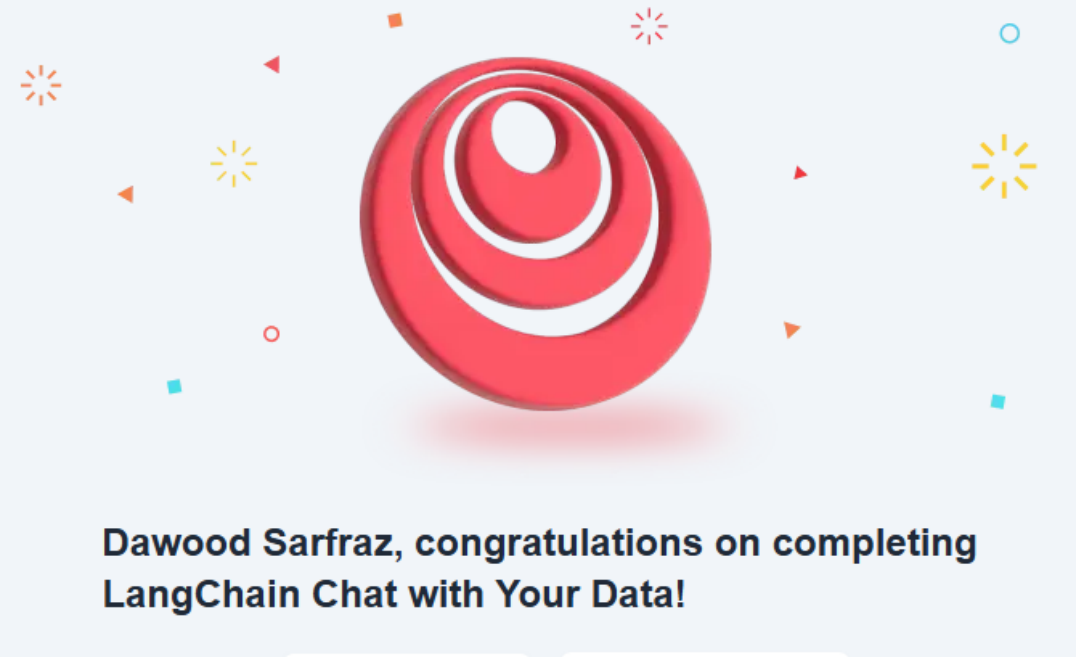

Figure 25: LangChain Chat with Your Data

Multimodal RAG Chat with Videos

Figure 26: Multimodal RAG Chat with Videos

Multimodal RAG Chat with Videos

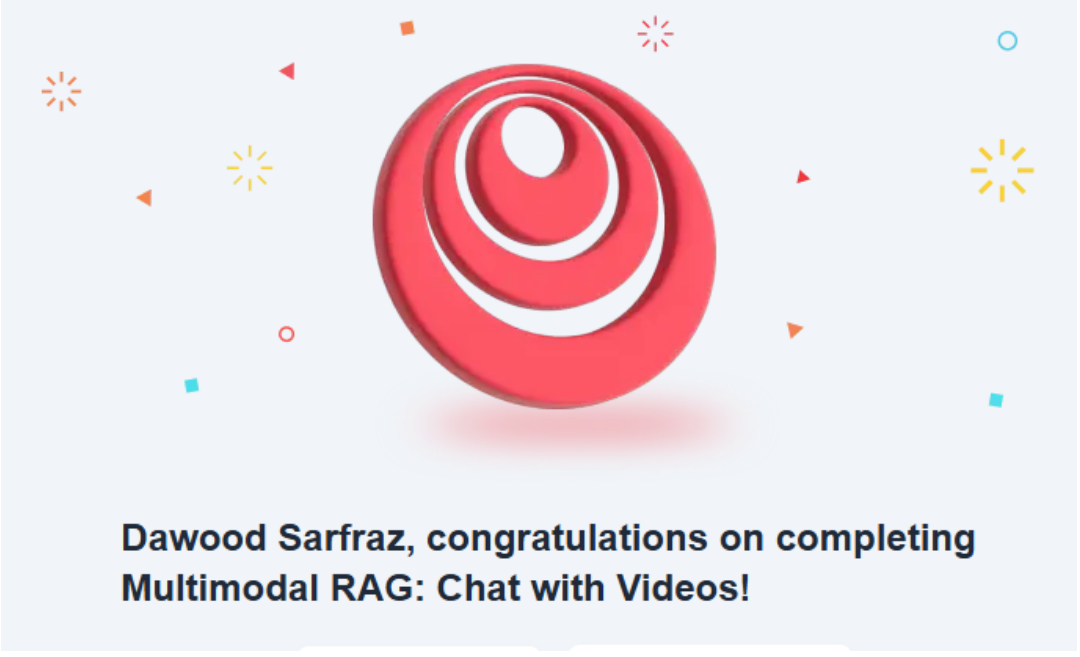

Figure 27: Multimodal RAG Chat with Videos

Finetuning Large Language Models

Figure 28: Finetuning Large Language Models

AI Agents in LangGraph

Figure 29: AI Agents in LangGraph

Building and Evaluating Advanced RAG

Figure 30: Building and Evaluating Advanced RAG

Building Applications with Vector Databases

Figure 31: Building Applications with Vector Databases

Building Multimodal Search and RAG

Figure 32: Building Multimodal Search and RAG

Prompt Engineering for Vision Models

Figure 33: Prompt Engineering for Vision Models

Prompt Engineering with Llama 2&3

Figure 34: Prompt Engineering with Llama 2&3

How Diffusion Models Work

Figure 35: How Diffusion Models Work

Introducing Multimodal Llama 3.2

Figure 36: Introducing Multimodal Llama 3.2

Vector Databases from Embeddings to Applications

Figure 37: Vector Databases from Embeddings to Applications

Event-Driven Agentic Document Workflows

Figure 38: Event-Driven Agentic Document Workflows

Function-calling and data extraction with LLMs

Figure 39: Function-calling and data extraction with LLMs

Functions, Tools and Agents with LangChain

Figure 40: Functions, Tools and Agents with LangChain

Knowledge Graphs for RAG

Figure 41: Knowledge Graphs for RAG

Large Language Models with Semantic Search

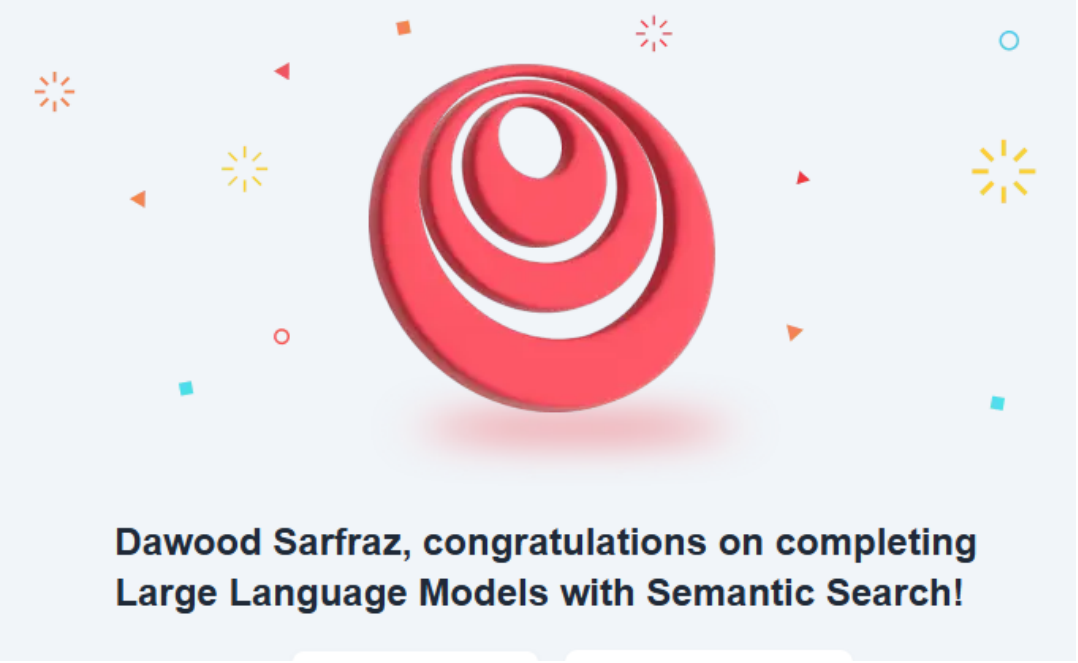

Figure 42: Large Language Models with Semantic Search

Retrieval Optimization Tokenization to Vector Quantization

Figure 43: Retrieval Optimization Tokenization to Vector Quantization

Reasoning with o1

Figure 44: Reasoning with o1

Intro to Federated Learning

Figure 45: Intro to Federated Learning

Credit Card Fraud Detection

Figure 46: Credit Card Fraud Detection

Building Recommendation Systems

Figure 47: Building Recommendation Systems