

# MANASVI KALYAN

+ (91) 9027723598 | [manasvikalyan63@gmail.com](mailto:manasvikalyan63@gmail.com) | [HuggingFace](#) | [leetcode](#) | [Github](#) | [Linkedln](#)

## Profile

B.Tech CSE student specializing in AI and Machine Learning, strong foundation in data structures, algorithms, and Python/C++. CGPA of 9.09 reflecting academic excellence. Passion for AI evident in projects, including Generative AI, NLP and Deep learning.

## Work Experience

**Quantstech** - AI/ML, Algorithmic Trading, NLP, Streamlit, Langchain, OpenAI API, Kite Connect, REST API Nov '23 – March'24

### AI/ML Python Developer Intern

- Task 1
- Developed a website featuring 4 features, Resume Parser utilizing Hugging Face libraries and OpenAI API, Job Description generator, Job based questions and answers generator and a Resume to Job Similarity calculator.
  - Deployed using Streamlit with Langchain and OpenAI API for conversational text generation with memory capabilities for 100% content retrieval.
- Task 2
- Created a tool with Streamlit and OpenAI API, which converts broker documents into the company's preferred format, engages in stock market analysis, filtering data for relevance and formatting. Visualization and calculates profit, loss, and drawdown within specified date ranges.
  - Deployed findings on Streamlit for user-friendly access with strategy generator that can analyze the data in 2/3 of time compared to manually.
  - Engineered data extraction and analysis of Nifty 500 dataset using YFinance library, and historical data from Zerodha via KiteConnect API.

## Projects

- RAG Enabled Research Chatbot** (Python, Langchain, Chainlit, RAG, NLP, Ollama, Chroma) June '24
  - Implemented a system using NLP and RAG technology integrated with ChainLit for conversational AI, utilizing LangChain for natural language processing and Ollama embeddings to transform research papers into vector representations.
  - User questions are processed to retrieve relevant information from a Chroma vector database, created by loading PDF, markdown, and text files, splitting documents into chunks, and converting them into embeddings using OllamaEmbeddings.
  - Utilized ChainLit and the Ollama model ("qwen:0.5b") to develop a responsive chatbot that processes incoming messages, manages conversation history for context, sends queries to the model for processing, returns responses, handles file loading, and maintains seamless conversation flow.
- LLM Fine-tuning and Model Deployment** (HuggingFace, PEFT techniques – Qlora, Transformers, LLMs, Data Modelling) May '24
  - Deployed models such as Llama-2-7b-chat-finetune and Finetune-Llama-3-8b-Alpaca using Hugging Face Spaces.
  - Fine-tuned Llama 2 on a multilingual dataset i.e. guanaco-llama2-1k using QLora techniques, for optimizing model's performance.
  - Achieved optimized performance of the deployed models by leveraging advanced fine-tuning strategies and deployment by HuggingFace Spaces.
- Flight Price Prediction** (Python, Scikit-learn, TensorFlow HuggingFace Spaces, Docker, Deep Learning, ML, HTML, CSS, Flask) Jan '24
  - Developed a machine learning model using RandomForestRegressor to predict flight prices based on various features such as airline, source, destination, departure and arrival times, and duration. Utilized data preprocessing, feature engineering, and hyperparameter tuning techniques to optimize the model's performance, achieving a high R-squared score for price prediction.
  - Bundled into a web application using Flask and the model is deployed as an API with UI made of HTML and CSS, Hosted on HuggingFace Spaces as a Docker Image.
- StoryGenAI** (Python, HuggingFace Transformers, NLP, Deep Learning, Streamlit, Language Models) May '24
  - Uploaded and preprocessed a storytelling dataset on Hugging Face, implementing NLP techniques for cleaning and preprocessing to enhance data quality and model performance.
  - Fine-tuned GPT- 2- medium model for less training time on storytelling and integrated streamlit for UI. The model takes input of few words and by prediction of next words it predicts the complete story. Also tested various models including LSTM, Transformers and BERT.
- Text Analysis App** (Python, HuggingFace Spaces, NLP, Flask, Deep Learning, Streamlit, T5, pypdf, NLTK) Dec '23
  - Developed a versatile text analysis app to calculate similarity scores (cosine similarity, BERT Score, and ROUGE Score) between the two pdf and highlighted the similar part. Additionally, it provides Text Summarization and Sentiment Analysis capabilities of pdf using pre-trained T5 models and NLTK's Vader sentiment analysis tool with visualization respectively.
  - Used Streamlit for user interface design and Deployed using Hugging Face Spaces. Also, tested using Flask API.
- Face Detection** (Python, TensorFlow, Deep Learning, OpenCV, Cascade Classifier) Aug '23
  - Developed a real-time face detection project using Python and OpenCV. Built and tested model with the CascadeClassifier from the OpenCV. Implemented a capture loop at 20 FPS, convert them to grayscale, and detect faces using the model.
  - Annotated the detected faces within rectangles with coordinates (x, y, w, h). Ensured proper resource release upon exiting and well displayed annotated frames for visualization.

## Achievements

- Selected for **Amazon ML Summer School '23** where I acquired knowledge about machine learning and deep learning including supervised learning, deep learning, reinforcement learning, probabilistic graphical models, sequential models, and the actual mathematics behind them.
- Solved **300+ LeetCode** DSA problems in C++.
- Actively contributed to the **HuggingFace** community by uploading datasets, models, and fostering collaboration and knowledge sharing.

## Publications

- Enhancing Legal Document Summarization for Professionals: An Extractive Approach** Accepted in 15<sup>th</sup> ICCCNT '24 – yet to be published
- (NLP, HuggingFace Transformers, TensorFlow, Machine Learning, Deep Learning, LLMs)
- Developed an extractive summarization system for legal documents by creating our own labeled datasets. Categorizing sentences based on their similarity to sentences in the corresponding summaries, using both binary (included/not included) and multi-class (Facts, Arguments, Issue, Analysis, None) labeling methods.
  - Fine-tuned models using a combination of traditional machine learning algorithms (Random Forest Classifier, Support Vector Machine), advanced neural networks (BiLSTM, Fully Connected Neural Network), and transformer-based models (BERT, RoBERTa) to evaluate performance metrics such as precision, recall, F1-score for the classification task and ROUGE, BERTScore for summaries generated.

## Skills

- Technical Skills:** DSA, Generative AI, Machine learning, Natural Language Processing, Deep Learning, Cloud Computing (AWS), Git, Docker
- Languages:** C++, Java, Python, SQL, HTML, CSS
- Framework:** Scikit-learn, TensorFlow, PyTorch, Streamlit, Langchain, Hugging Face, Flask, RAG, Chainlit, llama-index, Rest API
- Libraries:** Matplotlib, Pandas, Plotly, KerasCV, KerasNLP, NumPy, NLTK, Spacy, OpenCV
- Soft Skills:** Teamwork, Leadership, Research, Helpful

## Education

**Bennett University, Greater Noida** Sep '21 - Oct '25  
B. Tech CSE | Specialization: Artificial Intelligence (CGPA: 9.09)

## Certificates

- [IBM Machine Learning Professional Certificate](#)
- [Natural Language Processing with Classification and Vector Spaces](#)