RUTVIK CHAVHAN

Nagpur, India | 9021744216 | chauhanrutvik42@gmail.com | LinkedIn | Github

TECHNICAL SKILL

- **Data Science & Analytics:** Python, NumPy, Pandas, SQL, EDA, Statistics, Data Visualization, Power BI
- **Machine Learning & AI:** Scikit-learn, TensorFlow, Keras, CNN, ANN, RNN, OpenCV, Transfer Learning, Time Series
- **NLP & Generative AI:** NLTK, BERT, RAG, YOLOv8, LangChain, Prompt Engineering, LLMs
- **MLOps & Deployment:** MLflow, Docker, Databricks, Streamlit, Flask, APIs, CI/CD
- **Advanced Machine Learning:** Hyperparameter Tuning, Feature Engineering, Model Optimization, A/B Testing
- **Data Processing & Deployment:** Flask, Docker, Google Cloud AI

EDUCATION

Bachelor of Technology in Computer Technology 7.8 CGPA
Rashtrasant Tukadoji Maharaj Nagpur University **06/2020 – 06/2024**

CERTIFICATION

Data Science and A.I **10/2024 – 06/2025**
*Gained expertise in **machine learning, and data analytics**, with hands-on experience in Python, SQL, and cloud platforms.*

PROJECT

Investment Agent: Autonomous Investment Advisor using LangChain

Technologies: Python, LangChain, Google Gemini LLM, LangGraph (Pregel), Flask, HTML, Prompt Engineering

- Engineered an autonomous financial planning agent using **LangChain, Google Gemini LLM**, and LangGraph to generate personalized investment allocations based on user inputs (age, risk profile, goal).
- Designed a multi-step agent workflow with **Pregel-based state management** (LangGraph) to manage user input collection, LLM-based reasoning, and portfolio recommendation delivery.
- Integrated Google Gemini LLM via **langchain_google_genai** to simulate expert-level financial advice generation in natural language.
- Built and deployed a Flask-based web application with an HTML frontend to capture user input and present dynamic portfolio recommendations.
- Applied **prompt engineering** and modular LangChain tools to create reusable reasoning functions for asset allocation logic.

Movie Review: Sentiment Analysis through Fine-Tuned BERT Transformer Model

Technologies: Python, HuggingFace Transformers, BERT, PyTorch, Pandas, NumPy, Scikit-learn, Matplotlib

- Fine-tuned a **pretrained BERT model (`bert-base-uncased`)** using the HuggingFace Transformers library to classify movie reviews as positive or negative.
- Preprocessed text data with **tokenization, padding, attention masks**, and created a custom **PyTorch Dataset** for efficient batching and training.
- Utilized **AdamW optimizer**, **CrossEntropyLoss**, and **learning rate scheduling** to train the model and avoid overfitting.
- Achieved robust classification performance, demonstrating strong understanding of **transfer learning**, **NLP pipelines**, and **transformer-based models**.