

Kirti Sharma

✉ kskirtisharma2000@gmail.com

📍 India

🔗 github.com/Kirtisharma2000

☎ 9522622157

🌐 linkedin.com/in/kskirti-sharma

Professional Summary

Final-year B.Sc. CS student (CGPA 7.75) with a 9-month AI Minor from IIT Ropar specialising in end-to-end machine learning systems. Built three production-deployed ML projects from scratch — a spam classifier (97% accuracy, Voting Classifier), an emotion prediction system with XGBoost and uncertainty modelling, and a collaborative filtering recommender trained on 1.1M+ ratings. Strong in model selection, hyperparameter tuning, ablation studies, and Flask/Streamlit deployment. Currently deepening skills in LangChain and RAG pipelines.

Education

B.Sc. Computer Science, NIEM, Bhind 07/2024 – Present
CGPA 7.75 | Data Structures, Algorithms, DBMS, Operating Systems

Minor in Artificial Intelligence, IIT Ropar 08/2024 – 04/2025
9-month program: Machine Learning, Deep Learning, NLP, Computer Vision, statistical modelling, end-to-end AI system design, hyperparameter tuning, and deployment pipelines

Skills

ML & AI

Python, Scikit-learn, XGBoost, TensorFlow (learning), Statistical Modelling, Feature Engineering, Ensemble Learning, Hyperparameter Optimisation, Ablation Studies, Model Evaluation

Deployment

Flask, Streamlit, Git, GitHub, Jupyter Notebook, VS Code, pickle

Currently Learning

LangChain, RAG Pipelines

NLP

NLTK, TF-IDF (unigrams + bigrams), Text Classification, Sparse Matrix Stacking, Stopword Removal, Stemming

Data

Pandas, NumPy, SQL, Matplotlib, Seaborn, Data Wrangling, Data Cleaning

Certificates

- Python (Basic) — HackerRank
- SQL (Basic) — HackerRank
- Problem Solving (Basic) — HackerRank

Projects

[AI Emotion Tracker](https://github.com/Kirtisharma2000/AI_emotion_tracker) [🔗](#), github.com/Kirtisharma2000/AI_emotion_tracker

- Engineered unified feature set combining TF-IDF (5,000 features, bigrams) with scaled behavioural metadata via `scipy.sparse.hstack`, enabling richer statistical modelling than text-only approaches
- Applied Logistic Regression (GridSearchCV, 5-fold CV) to classify 6 emotional states; XGBRegressor (RandomizedSearchCV) predicts emotion intensity 1–5, translating output into actionable recommendations
- Implemented uncertainty modelling: confidence <0.55 flagged uncertain; <0.40 returns safe default — preventing overconfident predictions and improving system reliability
- Conducted 3-way ablation study (text-only, metadata-only, combined) quantifying each feature group's contribution; findings directly informed final architecture and feature selection decisions
- Deployed as Flask web application with rule-based decision engine mapping predicted emotion + intensity + context → specific user actions (box breathing, deep work, journaling, rest)

[SMS / Email Spam Classifier](https://github.com/Kirtisharma2000/sms-spam-classifier) [🔗](#), github.com/Kirtisharma2000/sms-spam-classifier

- Built full NLP pipeline on 5,572 SMS messages — tokenisation, stopword removal, stemming, TF-IDF vectorisation transforming raw text into structured, analysis-ready data
- Benchmarked SVM, Decision Tree, and Extra Trees; combined into Voting Classifier achieving 97% accuracy — outperformed every individual model, reducing false positives critical for spam filtering
- Deployed as live Streamlit web app; evaluated via accuracy, precision, and confusion matrix analysis

[Book Recommender System](https://github.com/Kirtisharma2000/Book_recommender_system) [🔗](#), github.com/Kirtisharma2000/Book_recommender_system

- Processed 1.1M+ ratings across 271,360 books and 278,858 users; filtered to 811 active users and 706 books to reduce sparsity and ensure statistically reliable recommendations
- Built user-item pivot table (706×810); cosine similarity returns top-5 personalised recommendations
- Engineered a two-engine system: popularity engine ranks top 50 books from 241,071 titles (min 250 ratings) for new users; collaborative engine handles personalised queries — all model artefacts serialised with pickle for efficient deployment
- Deployed via Flask backend with HTML/CSS frontend, delivering data-driven book recommendations through a clean user-facing web interface