

Subash Pandey

Date of birth: 14/08/2000 | **Email address:** axlesubash111@gmail.com | **Website:** https://www.subash-pandey.com/ | **LinkedIn:** subash-pandey-ai |

Address: Milan tole, Tokha-8, 44600, Kathmandu, Nepal (Home)

ABOUT ME

I'm an AI/ML engineer from Nepal with a strong interest in Generative AI and applied machine learning. I enjoy exploring the intersection of research and implementation, and I'm driven by curiosity, clarity, and continuous learning. I take pride in writing clean code, thinking deeply about problems, and building meaningful, responsible AI solutions that create real impact.

WORK EXPERIENCE

REMOTE AI/ML ENGINEER – SCOPIC SOFTWARE LLC – 23/10/2024 – Current – KATHMANDU, NEPAL

Website: https://scopicsoftware.com/

- Develop and maintain an AI Receptionist platform, implementing AI-driven features and resolving production bugs to sustain conversational quality and system reliability across multiple deployments.
- Lead development and deployment of a GenAI sales chatbot integrated with Zoho CRM, using LangGraph, RAG pipelines, and Qdrant to automate lead qualification, intent detection, user info extraction, lead scoring, and SEO tracking (referrer, session time, last visited page), processing 5+ warm leads a month.
- Conduct R&D on audio feature extraction (pitch, pause, pace) using Parselmouth, Librosa, and Praat to inform internal voice-based AI capabilities.
- Scope AI projects and delivery timelines by breaking technical tasks into phases with man-hour-level estimates, supporting both internal initiatives and client-facing projects.
- Maintain and expand the AI service layer of an interviewer platform, covering question generation, grading logic, prompt engineering, and end-to-end GenAI integration.
- Build evaluation pipelines to assess chatbot and RAG performance, improving response accuracy by 20% and lowering latency and cost through continuous grounding improvements.
- Collaborate with fully remote, cross-functional teams following Git best practices, agile workflows, and internal security policies; translate business goals from non-technical stakeholders into deliverable AI features.
- Contribute as a full-stack developer to a keyword research and SEO SaaS platform: build backend services, manage database schemas and migrations, integrate subscription billing, implement authentication flows, and develop AI-driven features for keyword analysis and content insights.
- **Tech stack:** Python, LangGraph, LangFuse, TypeScript, Node.js, FastAPI, Docker, AWS (Lambda, Fargate, OpenSearch), GCP (Cloud Run), Serverless, Redis, MySQL, CI/CD.

AI/ML ENGINEER – PEACE NEPAL DOT COM – 01/07/2024 – 30/11/2024 – KATHMANDU, NEPAL

Website: https://peacenepal.com/

- Designed and developed chatbots for domain-specific use cases including banking, travel, and customer support.
- Implemented Retrieval-Augmented Generation (RAG) and a multi-agent framework to improve response quality and relevance.
- Integrated databases and external knowledge sources to support comprehensive, grounded answers.
- Planned and managed chatbot development projects end-to-end, from requirements through deployment.
- Fine-tuned LLMs for domain-specific accuracy across 5 use cases.
- Conducted testing and optimization for accuracy and performance; maintained documentation and reported on project progress.
- Deployed interactive web chat applications for production use.

AI/ML ENGINEER – ICEBRKR AI SOLUTIONS. – 22/03/2024 – 14/06/2024 – KATHMANDU, NEPAL

- Engineered real-world AI solutions using machine learning to address applied business problems.
- Built and deployed scalable ML models for data analysis and automation in cloud environments.
- Collaborated with cross-functional data science and engineering teams to deliver production-ready solutions.
- Applied and evaluated frameworks including HuggingFace, PyTorch, and AWS to stay current with AI/ML tooling.

DATA ANALYST – CONTENTIO LAB PVT. LTD. – 31/08/2021 – 28/04/2022 – KATHMANDU, NEPAL

- Analysed large datasets using Python to identify trends and patterns, fast-tracked the process of data wrangling.
- Created clear and concise data visualisations (charts, dashboards) using Python libraries (e.g., Matplotlib, Seaborn) that effectively communicated insights to stakeholders.
- Developed and maintained robust data pipelines using Python scripting to ensure data accuracy and efficiency, leading to a 20% reduction in data processing time.
- Supported data-driven decision making across various departments (marketing, sales, etc.) by providing Python-powered analysis (e.g., customer segmentation, churn prediction), leading to cost-saving measures through market analysis.
- Proficient in data analysis tools (e.g., SQL, Python libraries like Pandas, NumPy) and comfortable working with large and complex datasets.

FRONTEND INTERN – IMARK PRIVATE LIMITED – 10/10/2020 – 25/01/2021 – KATHMANDU, NEPAL

- Developed a fully functional to-do web application using React.js, featuring functionalities like task creation, management, and completion.
- Implemented a CRUD (Create, Read, Update, Delete) backend to support data persistence for the to-do application, demonstrating understanding of data interaction.
- Integrated a secure login component using React for user authentication.

EDUCATION AND TRAINING

26/09/2022 – 17/12/2023 Exeter, United Kingdom

MASTER OF SCIENCE IN DATA SCIENCE University of Exeter

Main Subjects:

- Statistics and Data Modelling: Statistical Data Modelling, Fundamentals of Data Science, Stochastic Processes
- Machine Learning: Machine Learning, Learning from Data
- Social Network Analysis: Social Networks and Text Analysis
- Data Science Applications: Introduction to Data Science, Digital Business Models

Occupational Skills:

- Data Analysis: Ability to clean, manipulate, and analyze large datasets.
- Machine Learning: Ability to design, build, and deploy machine learning models.
- Data Visualization: Ability to create clear and informative visualizations of data.
- Communication: Ability to communicate complex data insights to both technical and non-technical audiences.
- Problem-solving: Ability to identify and solve problems using data-driven approaches.
- Business Acumen: Understanding of how data science can be applied to solve business problems and create new opportunities.

Website https://www.exeter.ac.uk/ |

Field of study Inter-disciplinary programmes and qualifications involving natural sciences, mathematics and statistics |

Final grade Merit | **Level in EQF** EQF level 7 | **National classification** 7 | **Type of credits** ECTS | **Number of credits** 90 |

Thesis Machine Learning the Steam Video Game Database

12/08/2019 – 10/12/2021 London, United Kingdom

BACHELOR OF SCIENCE (HONORS) COMPUTING London Metropolitan University

Main Subjects:

- Computer Science Fundamentals: Computer Hardware and Software Architectures, Fundamentals of Computing, Logic and Problem Solving, Programming
- Software Development: Advanced Programming and Technologies, Software Engineering, Application Development, Project
- Database Management: Databases
- Networking: Network Operating Systems
- Emerging Technologies: Cloud Computing and the Internet of Things, Mobile Application

Occupational Skills Gained:

- Programming: Proficiency in programming languages (Python and Java).
- Software Development: Ability to design, build, test, and deploy software applications.
- Problem-solving: Applying logic and programming skills to solve complex problems.
- Database Management: Understanding and managing databases for data storage and retrieval (MySQL).
- Networking Concepts: Knowledge of network infrastructure and operating systems.

- Project Management: Experience in planning, executing, and delivering a software development project.
- Professionalism and Ethics: Understanding of professional conduct and ethical considerations in the computing field.

Website https://www.londonmet.ac.uk/ | **Field of study** Field unknown | **Final grade** 2:1 |

Thesis Floki - An eCommerce Web Application

PROJECTS

Craft - Claude Code Skills Marketplace

A cross-platform plugin marketplace and installer CLI for Claude Code. It packages my own agent skills into bundles alongside third-party skills that I reviewed line by line and pinned to specific commits. It covers the whole workflow, from writing a skill to evaluating it and releasing it.

- CLI Tooling: Built skills, a dependency-free Node.js CLI with an interactive picker, scenario recipes (skills use existing-repo), and add/remove/sync/doctor/self-update commands. It installs through Claude Code's native plugin system and never copies files into a project.
- Scoped Installs & Lockfiles: Designed three install scopes: shared with the repo, just me, and global. Each scope has its own settings file and lockfile. "Just me" files are git-excluded automatically, so client and open-source repos stay clean, and a fresh clone or new machine restores everything with one skills sync.
- Agent Skills: Wrote 17 skills across 5 dependency-free plugins: engineering workflow (PR authoring, security gate, ADRs, test integrity, docs), ops (incident response, bug triage, observability, dependency upgrades, releases), UI, open-source contribution, and writing.
- Guardrail Hooks: Built Claude Code hooks for three jobs. A push guard warns or blocks when a branch touches sensitive files (auth, secrets, CI, infra, lockfiles) and has no security review on record. A test guard catches edits that weaken tests, such as .skip, deleted assertions, lowered coverage or --no-verify. A first-prompt hook deepens the opening request. All three are set through environment variables (warn/enforce/off).
- Supply-Chain Review: Pinned 42 third-party skills to audited commits, with a written review for each. Safer settings are applied on install and removed on uninstall: telemetry off, Sonnet-capped review models, and a deny-list for browser automation. Plugins that run unreviewable binaries are opt-in only.
- Evaluation: Gave every skill 7 eval cases (3 that should trigger, 3 near-misses, 1 graded task) run with claude plugin eval and an LLM judge. A generated report shows a 100% trigger and no-trigger rate on the skills evaluated, and task-score gains up to +0.67 over the no-skill baseline. CI fails if a skill has no eval coverage.
- CI/CD & Releases: GitHub Actions runs on Windows, macOS and Linux and covers install/uninstall round trips (checking git status is clean afterwards), hook behaviour, shell completion, and version checks. Releases follow semver with tags and a changelog, and the CLI checks once a day for a newer release.
- Developer Experience: One-command setup (npm i -g then skills) with no manual PATH edits, tab completion for bash, zsh, fish and PowerShell that installs itself, and a playbook that maps situations to the skills to use.

Link https://github.com/notsubash/claude-skills

Cloud Native AI Platform

A production-oriented, cost-bounded platform serving a minimal AI summarization API, demonstrating end-to-end cloud-native engineering from local development through GitOps deployment.

- API Development: Built a FastAPI service with health/readiness probes, Prometheus metrics endpoints, and an LLM abstraction layer supporting both stubbed testing and live inference (DeepSeek via OpenAI-compatible API).
- Containerization: Designed a multi-stage Dockerfile with non-root runtime, minimizing image size and attack surface.
- Infrastructure as Code: Provisioned a Hetzner VPS and k3s cluster using Terraform, with IP-locked firewall rules and cloud-init bootstrapping.
- Kubernetes & Helm: Authored Kubernetes manifests (Kustomize) and a Helm chart with environment-specific values (local/hobby), supporting upgrades and rollbacks.
- CI/CD: Implemented a GitHub Actions pipeline (test → build → push) publishing immutable images to GHCR, with Helm lint/template validation on every PR.
- GitOps: Deployed Argo CD to manage declarative, Git-driven releases to the cluster, eliminating manual helm upgrade from developer machines.
- Observability: Set up kube-prometheus-stack, Loki, and Promtail for metrics, logging, and alerting, with a custom RED (Rate/Errors/Duration) dashboard and request-ID log correlation.
- Cost Engineering: Designed pause/resume workflows (VPS power-off vs. Terraform destroy) to keep total infrastructure spend under $15/month.
- Roadmap (in progress): Extending the platform with autoscaling (HPA/KEDA), canary deployments via Argo Rollouts, automated backup/restore drills, multi-environment promotion, and CI security hardening (Trivy scanning, RBAC least-privilege).

Link https://github.com/notsubash/cloud-native-AI-platform

30/10/2026 – 13/11/2026

LastCall

A web app that watches the links in scheduled social posts and holds a post before it publishes something that's no longer true. Built solo on the Buffer API for the Buffer Commit contest, from spec to production, with safety rules written down up front and enforced by tests.

- Product: Signs in with Buffer OAuth and reads the user's queue. It checks every link in it (sold-out products, closed jobs, cancelled events, removed videos, dead pages). Stale posts become Buffer drafts and go back to their original time when the link is fresh again. Every decision shows the evidence behind it.
- Deterministic Verdict Engine: A rule engine made of pure functions, with no AI and no scoring. Checkers only collect evidence, and the engine alone decides Fresh, Stale or Can't check. A blocked or unclear page can never cause a hold.
- Link Checkers: Parsers for Shopify and WooCommerce stock, schema.org JSON-LD, Greenhouse/Lever/Ashby job boards, YouTube oEmbed, link cards, media reachability and HTTP status. All sit behind a polite fetcher: identified User-Agent, 1 request per domain per 5 s, timeouts, size caps, Retry-After, and no attempt to get around blocks.
- Accuracy Harness: A set of 150 real public URLs with known answers, run through the real fetcher, checkers and engine. It reached 100% Stale precision (zero false holds) and 95% recall. The build fails if precision drops below 100%.
- API Safety & Rate Budgeting: All Buffer GraphQL traffic goes through one gateway with a request ledger and budget governor, and batches aliased queries. This keeps a rate limit shared by every user to about 12 requests per user per day. Each write is logged before it's sent and checked afterwards. The app never publishes, and it never changes post content on its own.
- Token Security: OAuth with PKCE, tokens encrypted at rest with AES-256-GCM, and one owner per token for single-use refresh-token rotation, with version checks so a reused token can't revoke the whole grant.
- Stack & Deployment: TypeScript, Hono, SQLite and React (demo only, compiled at build time), deployed with Docker Compose and Caddy (automatic HTTPS, CSP, HSTS) on a Hetzner VPS. Includes a live demo shop for testing.
- Testing: Test-driven throughout, with 1,773 Vitest tests across 72 files and table-driven tests for the core logic. Includes a mock token server that copies real token rotation, Playwright end-to-end tests with axe accessibility checks, and 13 ADRs (architecture decision records).
- Design: Server-rendered pages with almost no JavaScript, styled as a departures board (split-flap status changes, dark and light themes). Also has a 30-day calendar, insights, a status page and email alerts.

Link https://lastcall.subash-pandey.com

Latent Parking Planner

A diagnostics-first parking planner that imagines short rollouts in latent space, plans with CEM MPC, and gates maneuvers when imagination looks untrustworthy, demonstrating end-to-end model-based control from simulation through learned world models and honest failure measurement.

- Simulation Loop: Built on Farama highway-env parking-v0 with Gymnasium, modeling goal-conditioned parking as continuous throttle/steering control on bicycle kinematics.
- True-Dynamics Planning: Implemented receding-horizon CEM MPC and validated successful parking on empty and neighbor-filled lots using real env.step imagination.
- Learned Dynamics & JEPA: Trained next-state MLPs and a JEPA-style latent predictor (encoder, EMA target, predictor in z), benchmarked against identity and open-loop horizon error.
- Latent MPC: Planned toward goal embeddings with batched CEM; compared receding, open-loop, and random policies across seeds.
- Distrust Gate: Instrumented one-step residual, ensemble disagreement, abort logic, and precision/recall vs. chance for pre-crash detection.
- Artifacts: Authored 8 browser lessons, reproducible scripts, and tracked plots/gifs; published a full experiment write-up.
- Results: True-dynamics CEM parks; learned latent MPC does not (0/6 seeds); distrust signals sit near chance for catching late crashes.

Links https://github.com/notsubash/latent-parking | https://www.subash-pandey.com/blog/when-a-self-parking-car-cannot-trust-its-imagination

01/06/2026 – CURRENT

Gavel

This project explored the use of multi-agent AI systems for critiquing startup ideas by simulating a panel of domain-expert personas. The project aimed to deliver structured, realistic, and balanced feedback that mirrors real-world evaluation from investors, engineers, and end users.

- Concept & Architecture Design: Designed a multi-agent pipeline where five independent AI judges: Venture Capitalist, Engineer, Product Manager, Customer, and Competitor; each evaluate a startup idea from their unique perspective.
- Agent Development: Built individual sub-agents using LangChain and Ollama (local LLMs), with Pydantic schemas ensuring each agent returns structured verdicts including scores, key concerns, and recommendations.
- Debate Pipeline: Implemented a LangGraph powered multi-round debate mechanism where agents argue and refine their positions across configurable rounds before reaching conclusions.
- Moderator Synthesis: Developed a moderator agent that aggregates all judge outputs and produces a final overall verdict summarizing the panel's consensus.
- Appeal Mode: Introduced a founder rebuttal feature allowing users to argue back against judge verdicts, triggering a follow-up evaluation round that accounts for the founder's counter-arguments.
- Data Persistence: Integrated SQLite-backed memory to store and retrieve past startup ideas and their evaluations for comparison and reference.
- Visualization & Export: Generated radar charts to visually represent judge scores across dimensions, with support for exporting full debate transcripts as Markdown.
- Frontend: Built an interactive Streamlit interface enabling users to submit ideas, follow live agent debates, and review structured results in real time.

Link https://github.com/notsubash/Gavel

CURRENT

The Underlying Curve

A code-first education project that turns AI and machine-learning ideas into silent, branded animations for YouTube, Instagram, and TikTok, from visual design through finished video and post-ready captions.

- Reach: 100K views and 1000+ followers in the first 40 days of posting on Instagram.
- Animation: Built reusable math scenes in Manim and rendered each episode twice in 4K, widescreen and vertical, instead of cropping one video for Shorts.
- Production: Assembled each video with FFmpeg (branding, music, sound hits) and generated captions, titles, and tags for each platform from one write-up

Links https://www.instagram.com/theunderlyingcurve/?hl=en | https://www.youtube.com/@theunderlyingcurve

Activity Recognition using Accelerometer and Gyroscope Data

Rebuilt a phone/watch IMU classifier from a leaky student notebook into a subject-independent pipeline on 51 people and 18 activities (accelerometer + gyroscope). The archived notebook's 85.6% phone accuracy came from shuffled overlapping windows with the same people in train and test. On the same flattened 5 s phone windows, leaky split macro-F1 is 0.89; 5-fold GroupKFold on subjects drops it to 0.29. The honest number is watch statistical XGBoost: 0.70 macro-F1 / 0.70 accuracy.

- Data and features. Audited the official UCI dump (15.6M rows). Interpolated sessions onto a shared 20 Hz grid, aligned accel and gyro, and built session-safe 5 s / 1 s windows. Default input is 104-d statistical features, not flattened raw samples.
- Models and eval. Compared dummy, logreg, random forest, and XGBoost. Primary metric is macro-F1. Phone XGBoost is 0.33 under GroupKFold (locomotion as a group is strong; eating and sitting are not). Watch on the dominant hand is the device that actually classifies the 18-way set. Tracked runs in MLflow; exported the watch head to ONNX and served it with FastAPI.
- Activities (18). Walking, jogging, stairs, sitting, standing, typing, brushing teeth, eating soup / chips / pasta / sandwich, drinking, kicking, catch, dribbling, writing, clapping, folding clothes.

Link https://github.com/notsubash/Activity-Recognition

01/03/2023 – 25/07/2023

Machine Learning the Steam Video Game Database

This project utilised network analysis to explore the evolution of game genres on the Steam platform. Key findings included:

- Emergent Sub-genres: Identified sub-genres evolving from popular categories like "Action," "Indie," and "Racing," highlighting genre diversification within the gaming market.
- Influential Genres: Pinpointed "Indie" and "Casual" genres as significant influencers within the Steam network, suggesting their impact on broader genre trends.

- Player Engagement Drivers: Revealed game ratings, release year, in-game achievements, and lower prices/free-to-play models as key factors driving player engagement.
- Societal and Economic Impact: Examined how genres like "Massively Multiplayer" and "Free to Play" contribute to cultural exchange, communication, and the overall growth of the gaming industry.

Links https://github.com/notsubash/Game-Genre-Network-Analysis | https://www.youtube.com/watch?v=IjWUKT-kXw4&t=23s

SKILLS

Data Science

Graph Networks | Data Science and Analytics | Machine Learning | Gephi | MLOps | Pandas | PyTorch | numpy

Presentation and Prototyping

Microsoft Office | Gradio | Tableau | Manim | Streamlit | Ffmpeg

Software Development

Python | Project Management | FastAPI | Stripe | TypeScript | Git | Audio Feature Extraction | Node.js

Cloud Computing

AWS | GCP | Docker | Serverless | Kubernetes | Helm | Argo CD | Infrastructure as Code (Terraform) | Prometheus, Grafana, Loki

LLM / AI

Langchain | LLM | Vector Database | LangGraph | NLP | LLM Finetuning | Prompt Engineering | GenAI Evaluation | LLMOps | Ollama

Databases

Databases | SQL | Redis | MySQL | Postgres | NoSQL

LANGUAGE SKILLS

Mother tongue(s): **NEPALI**

Other language(s):

	UNDERSTANDING		SPEAKING		WRITING
	Listening	Reading	Spoken production	Spoken interaction	
ENGLISH	C1	C1	C1	C1	C1
HINDI	C2	B2	B2	B2	B2

Levels: A1 and A2: Basic user; B1 and B2: Independent user; C1 and C2: Proficient user

---

I am eager to leverage my skills and experience to contribute to a team-oriented environment. I am a highly motivated individual with a strong desire to learn and grow. I am confident that I can make a significant impact on your organisation.