

# KUNAL SACHDEV

☎ 437-663-6032 ✉ Email 📁 Portfolio 🌐 LinkedIn 🐙 GitHub 🏆 Certifications

## EDUCATION

### University of Waterloo

Bachelor of Computer Science, Minors in Statistics and Economics

Sep. 2023 – Present

Waterloo, ON

## TECHNICAL SKILLS

**Languages/Cloud:** Python, C++, JavaScript, AWS, Azure, IBM Cloud, SQL, HTML/CSS

**ML/AI:** Pandas, NumPy, Scikit-learn, PyTorch, TensorFlow, Streamlit, LangChain, BeautifulSoup, Matplotlib, Plotly

**Web Development:** React, Next.js, Express.js, Node.js, Flask, Django, SQLAlchemy

**Databases:** MySQL, SQLite, MongoDB, Db2, Oracle

**Technologies:** Git, Docker, Kubernetes, Linux, bash, GDB, Confluence

## EXPERIENCE

### Full Stack Software Developer - Cloud and ML Co-op

Nokia Canada

January - April 2026

Kanata, ON

- Incoming - Cloud and Network Services (CNS) team. AI/ML solutions for global networks.

### Data Scientist Co-op - Full-Stack AI Software Development

Ontario Lottery and Gaming Corporation

Sept 2025 – Present

Toronto, ON

- Built a conversational AI platform using **React**, **Node.js**, **Python FastAPI**, and **Azure** to automate request onboarding using OpenAI, Jira, and Microsoft Graph **APIs**.
- Deployed a **multimodal RAG** chatbot with **LangChain** and **Azure AI Search** to query insights from business documents, saving **215+** hours/month for business teams.
- Fine-tuned an **XGBoost** classifier on **115M+** records using **Databricks**, **Hadoop**, and **PySpark** for segmentation, enabling targeted ad campaigns to increase customer engagement.

### GenAI Engineer Intern

Edelweiss Life Insurance

May 2025 – August 2025

Mumbai, India

GenAI-powered Business Insights with Visualization

- Built a serverless **RAG** app on **AWS Bedrock** enabling real-time conversational access to AWS Data Lake, replacing **200+** static BI dashboards and saving **Rs. 300,000+** annually.
- Fine-tuned CodeLlama-7B-Instruct using **QLoRA**; deployed to AWS **SageMaker** with **TensorRT-LLM**, accelerating inference by **1.8x** and cutting memory usage by **50%**.
- Engineered **Lambda** functions for query validation, **Redshift** schema extraction, and LLM inference; achieved **99%** API uptime and reduced **8,700+** annual emails per user.
- Enhanced SQL generation accuracy by **28%** using **Cohere** embeddings + **OpenSearch** retrieval, supporting **1.5K+** daily active users via real-time **Streamlit** UI.

Data Science – Policy Grievance & Mis-selling Prediction

- Productionized a **LightGBM** + **CATBoost** ensemble to flag high-risk policies across **809K+** policies (**96.9%** of grievance surfaced in top decile), enabling proactive intervention.
- Built a scalable ML pipeline on AWS **SageMaker**, containerized with **Docker**, and automated weekly predictions with **EventBridge**, reducing prediction latency by **66.7%**.

## PROJECTS

**Car Dealership App** | [Architecture](#), Python, Node.js, Express, MongoDB, Flask, Django, React, Kubernetes | 🐙

- Architected a full-stack microservices app with a **React** frontend, and **3** services: **Django** + **SQLite** backend, reviews REST APIs with **Express.js** + **MongoDB**, and a sentiment analyzer on IBM Cloud; CI/CD with GitHub, & deployment on **Kubernetes**.

**Winning the Space Race with Data Science** | BeautifulSoup, Db2, Matplotlib, Plotly, Scikit-learn | 🐙

- Developed an ML pipeline predicting Falcon 9 landing success with 87.5% accuracy, using GridSearchCV-tuned Decision Tree; automated EDA of 90+ launches via Db2, SpaceX API, BeautifulSoup, and designed Plotly/Folium dashboards.

**Biquadris** | [UML Diagram](#), C++, XWindows, Git, Cygwin, Bash, Linux, GNU, GCC, GDB | 🐙

- Implemented a two-player Tetris-style game in **C++** using modular **OOP**, applying **Factory** and **Observer** design patterns to support **6+** blocks, **3+** levels, **3** special actions, and real-time text/graphical rendering via XWindows.

**E-Commerce Plant Nursery** | React, Redux Toolkit, Vite | 🐙

- Built a responsive e-commerce plant nursery web application using **React**, **Redux Toolkit**, and **Vite**, leveraging React Hooks (**useState**, **useSelector**, **useDispatch**) for global state management, dynamic product listings, and state-driven UI transitions.