Akashdeep Singh – Data Science & Optimization Portfolio

📧 **Email:** imakashrandhawa@gmail.com
📱 **Phone:** 647-615-0058
🔗 **LinkedIn:** linkedin.com/in/imakashrandhawa
💻 **GitHub:** github.com/imakashrandhawa

1. AI-Powered Budget Advisor (Forecasting & Optimization) *(Ongoing)*

Objective: Build a real-time financial tracking and forecasting system capable of delivering actionable spending insights.
Technologies: Python, Flask, Pandas, NumPy, scikit-learn, AWS EC2, AWS RDS (MySQL), Docker, Kubernetes, GitHub Actions (CI/CD).
Key Contributions:

- Designed ML pipeline for transaction classification and budget forecasting using regression models.
- Built ETL process to ingest and clean transaction data from **Plaid API**.
- Deployed scalable backend on AWS with **Docker** containers orchestrated via **Kubernetes**.
- Integrated CI/CD workflows to automate testing and deployments.
  Outcome: Produced accurate monthly spending forecasts and trend analysis for early-stage testing.

2. CNN Image Classification Model *(2025)*

Objective: Classify images into multiple categories for potential use in automated quality control.
Technologies: Python, TensorFlow, Keras, scikit-learn, Pandas, NumPy.
Key Contributions:

- Implemented CNN architecture with hyperparameter tuning to improve classification accuracy.
- Applied data augmentation and dropout to enhance model generalization.
- Exported trained model as REST API for integration into production systems.
  Outcome: Achieved high validation accuracy on benchmark datasets and demonstrated readiness for real-world deployment.

3. Floyd-Warshall Optimization for Large Datasets *(2024)*

Objective: Implement and benchmark shortest-path algorithms for use in resource allocation and routing problems.
Technologies: Python, Docker, Bash scripting.
Key Contributions:

- Developed and optimized Floyd-Warshall algorithm for large-scale graph datasets.
- Created containerized testing environments for reproducibility and performance tracking.
- Automated multiple runs with shell scripts for benchmarking.
  Outcome: Reduced computation time for large input sizes, demonstrating viability for supply chain routing applications.

4. Automated Parking System (IoT + Cloud) *(University Project)*

Objective: Optimize parking slot allocation through real-time monitoring.
Technologies: Python, Firebase, IoT sensors, Data Processing Pipelines.
Key Contributions:

- Designed backend logic to allocate slots dynamically based on live IoT sensor data.
- Built data processing workflow to handle continuous streaming updates.
  Outcome: Reduced wait times and improved slot utilization efficiency in prototype testing.

🗁 Additional Work & Source Code

Full code repositories and additional projects available at:
github.com/imakashrandhawa