

# RUCHIT DOBARIYA

+1(438) 926-6504 ♦ Montreal, QC, Canada

[ruchitdobariya307@gmail.com](mailto:ruchitdobariya307@gmail.com) ♦ [LinkedIn](#) ♦ [ruchitdobariya.github.io](https://ruchitdobariya.github.io) ♦ [Github](#)

## EDUCATION

Concordia University, Montreal Master of Applied Computer Science	(September 2022 - Present)
Gujarat Technological University Bachelor of Computer Science	(July 2018 - May 2022)

## SKILLS

**Programming:** Python, Java, C/C++, Javascript  
**FrameWorks / Operating Systems:** Flask, FastApi, Django, TensorFlow, PyTorch, Linux  
**Database Management:** InfluxDb, SQL(MySQL), NoSQL  
**DevOps:** Docker, Kubernetes, Kubeflow, Google Cloud Platform, Camunda  
**Other Tools:** Git

## EXPERIENCE

Ericsson Machine Learning Intern	(September 2023 - Present) <i>Montreal, Canada</i>
<ul style="list-style-type: none"><li>Implementing <b>Multi-Agent Framework</b> in <b>Python</b> and <b>Java</b>, as well as engineering Data Parsers and Database Schema to <b>optimize</b> 5G network efficiency and reduce Query response time.</li><li>Integrating <b>Camunda</b> workflows to streamline processes and reduce operational bottlenecks.</li><li><b>Researching</b> and actively contributing to the implementation of <b>ML algorithms</b>, resulting in an <b>improvement</b> in predictive analytics accuracy and contributing to a enhancement in overall network performance.</li><li><b>Participating</b> in daily stand-ups to provide updates on project progress, discuss challenges, and collaborate with senior developers. Actively contributing to design processes by gathering requirements and collaborating with the development team.</li></ul>	
Orena Solutions Machine Learning Intern	(January 2022 - April 2022) <i>Vadodara, India</i>
<ul style="list-style-type: none"><li><b>Developed</b> a CNN model utilizing <b>Transfer Learning</b> and <b>Data Augmentation Techniques</b>, achieving <b>92.54% accuracy</b> in <b>Brain Tumor Classification</b>.</li><li><b>Optimized hyperparameters</b> and <b>evaluated model performance</b>, leading to improved accuracy and <b>robustness</b>.</li><li>Engineered end-to-end automated machine learning workflows utilizing <b>Git version control</b>, resulting in a <b>40% reduction in development time</b> and a <b>20% improvement in code quality</b>.</li></ul>	

## PROJECTS

Kubeflow-GNN - Python, PyTorch, Kubeflow ( <a href="#">github</a> )
<ul style="list-style-type: none"><li><b>Utilized SAGEConv</b> to perform <b>link property prediction</b> in <b>documents citation network data (ogbl-citation2)</b>, achieving an accuracy of <b>87.6%</b>.</li><li><b>Deployed GNN model Training</b> as <b>PytorchJob</b> in <b>Kubeflow</b>, which implements Pytorch training operator, resulting in a <b>20% reduction in training time</b>.</li><li><b>Implemented DDP (DistributedDataParallel)</b> for <b>Distributed Training</b> of the model, measuring accuracy and training time with <b>different epochs</b> (e.g., 50) and <b>number of workers</b> (e.g., 4), and observed a <b>12% increase in accuracy</b> with 4 workers.</li></ul>
Analysis of First Fit and CBIP Algorithms on Online Graph Coloring ( <a href="#">github</a> )
<ul style="list-style-type: none"><li>Designed and developed a <b>React Application</b> to analyze and compare the performance of <b>algorithms</b> for <b>Online Graph Colouring</b>.</li><li>Executed <b>algorithms</b> in <b>JavaScript</b> to colour the nodes of an <b>online graph</b> as they arrive in <b>real-time</b>.</li><li>Conducted <b>experiments</b> to evaluate the efficiency of <b>algorithms</b> on different types of graphs, including <b>random</b>, <b>Erdős-Rényi</b>, and <b>scale-free graphs</b>.</li></ul>
Two Phase Multiway Merge Sort (TPMWMS) - Java ( <a href="#">github</a> )
<ul style="list-style-type: none"><li>Utilized the <b>TPMWMS algorithm</b> to <b>partition and sort input files</b> in <b>main memory-sized runs</b>, and <b>wrote the runs back to disk</b>.</li><li>Merged the <b>sorted runs</b> from each file in the second phase using a multiway merge algorithm that minimized disk I/O.</li><li>Optimized <b>sorting and merging</b> by caching frequently accessed blocks in a <b>buffer manager</b> and <b>fine-tuned program parameters</b> using <b>profiling tools and parallel processing</b>.</li><li>Conducted <b>optimizations</b> including experimenting with <b>run size</b> (e.g., 1000), <b>merge passes</b> (e.g., 5), and <b>buffer size</b> (e.g., 512 KB).</li></ul>
Blog Web App - Python, Flask ( <a href="#">github</a> )
<ul style="list-style-type: none"><li>Utilized <b>Flask</b> framework to build the backend of the application, ensuring a lightweight and modular structure.</li><li>Employed HTML, CSS, and Jinja2 templating for creating a responsive and visually appealing user interface.</li><li>Integrated a relational database <b>PostgreSQL</b> for efficient data storage and retrieval.</li></ul>