RITAYAN PATRA

rp3247@columbia.edu | linkedin.com/in/ritayanpatra | github.com/RITS98 | rits98.github.io

EDUCATION

Columbia University New York City, NY
Masters of Science in Data Science Expected Dec 2025
Cumulative GPA: 3.6/4.0
Relevant Coursework: Machine Learning, Deep Learning; Computer Systems, Big Data Analytics, Probability and Statistics

Vellore Institute of Technology Vellore, India
Bachelors of Technology in Electronics and Communication Engineering Jul 2018 - Jun 2022
Cumulative GPA: 3.9/4.0
Relevant Coursework: Data Structure and Algorithms, Databases

SKILLS

Programming Languages: Python, Java, SQL
Libraries/Frameworks: NumPy, Pandas, Scikit-learn, Matplotlib, Seaborn, BeautifulSoup, PySpark, PyTorch, Ray
Tools / Platforms: AWS, GCP, Oracle Integration Cloud, Airflow, Hadoop, Hive, Spark, Kafka, Git, GitLab CI/CD, Docker, MLflow
Databases: PostgreSQL

WORK EXPERIENCE

Columbia Engineering New York, NY
Research Assistant May 2025 – Jun 2025

- Developed and deployed an NLP pipeline on Columbia's high-performance computing cluster to analyze the impact of 10,000 medical trials on surgical research
- Preprocessed and filtered 38M+ PubMed articles using BM25 to identify semantically related biomedical literature
- Generated dense embeddings with PubMedBERT and indexed them using FAISS for scalable and efficient similarity-based retrieval
- Built a lightweight Python interface for searching and retrieving related publications based on text similarity
- Provided weekly updates to faculty, ensuring progress aligned with research objectives and code was reproducible

NatWest Group Gurugram, India
Data Engineer Jul 2022 – Jul 2024

- Orchestrated and maintained data pipelines on Oracle Integration Cloud to streamline transaction and reference data processing of the bank enhancing operational efficiency for downstream teams
- Developed and deployed new ETL data pipelines to migrate legacy Oracle Hyperion Server to Oracle Integration Cloud, saving nearly 1 million British pounds annually on maintenance expenses of legacy systems
- Automated data staging and transformation processes using SQL and Python, enabling seamless Oracle Database and Oracle ERP integration
- Designed and implemented a test automation suite using Java and Selenium for Oracle ERP systems, slashing manual effort by 80% and reducing testing time from two weeks to three days
- Implemented GitLab CI/CD pipelines with senior software engineers for smooth deployment of code and artifacts to SIT, UAT, and Production environments

Nuclei Technologies Navi Mumbai, India
Data Science Intern Sep 2020 – Nov 2020

- Acquired proficiency in R, Python, and data analysis libraries like Pandas, NumPy, Matplotlib, and Seaborn
- Analyzed employee attrition and hypermarket data using Pandas, NumPy, and Matplotlib to uncover key insights

PROJECTS

NEAR REALTIME STOCK FORECASTING

Docker, Python, Airflow, AWS, Mlflow, Pytorch, Ray, Streamlit

- Designed and implemented a near real-time stock data ingestion pipeline using Airflow, PostgreSQL and AWS (DynamoDB Streams, Kinesis, Firehose, S3), achieving low latency for processing high-frequency financial data
- Developed and trained a hybrid CNN-BiLSTM model in PyTorch Lightning for time series forecasting; leveraged Ray for distributed training and integrated MLflow and DagsHub for end-to-end experiment tracking and model management
- Architected a scalable, partitioned data lake on AWS S3 using Apache Hudi and AWS Glue; automated metadata cataloging and optimized query performance with Athena and Glue Crawlers
- Deployed interactive dashboards with Streamlit and AWS QuickSight for stock price prediction and analysis

REDDIT API DATA PIPELINE

Docker, Airflow, Python, AWS

- Designed an end-to-end Reddit data pipeline processing 1,000+ posts using Apache Airflow, Python, and AWS cloud services (S3, Glue, Athena)
- Automated scalable ETL workflows with modular components for extraction, transformation, and loading, including schema enforcement
- Built a cloud-native data lake on AWS S3, integrated with AWS Glue Crawlers and Data Catalog for automated schema discovery and SQL querying via Athena
- Containerized the pipeline infrastructure using Docker Compose, orchestrating Airflow, PostgreSQL, and Python apps for consistent local and cloud deployment

IOT VEHICLE DATA ENGINEERING AND ANALYSIS

Docker, Kafka, Spark, AWS

- Built a mock real-time IoT data pipeline using Apache Kafka and Apache Spark Structured Streaming, processing 5 concurrent data streams (vehicle, GPS, traffic, weather, emergency)
- Designed and deployed a scalable architecture with Docker Compose, orchestrating containerized Kafka-Zookeeper and multi-node Spark clusters with health checks and monitoring via Spark UI
- Implemented a cloud-native data lake and warehouse solution on AWS, leveraging S3 (Parquet format), Glue (schema registry), and Redshift external tables with IAM-based access control
- Automated data quality enforcement and schema evolution, using glue crawlers, and optimized Parquet storage for low-latency querying and BI integration

CUSTOMER CHURN PREDICTION USING DEEP LEARNING AND MLOPS

Pythin, Pytorch, Mlflow, Optuna, Streamlit

- Developed an ANN-based churn prediction model using PyTorch Lightning with feature scaling, label/one-hot encoding, and automated hyperparameter tuning via Optuna
- Integrated MLflow for experiment tracking and artifact logging (ONNX model, scalers, datasets) and enabled streamlined model management and reproducibility
- Deployed interactive Streamlit UI for real-time churn prediction, with ONNX-exported model supporting portable and production-ready inference

URL SUMMARIZER USING LANGCHAIN AND GROQ API

Python, Langchain, Groq API, Streamlit

- Developed a Streamlit-based web summarization app using LangChain, Groq (LLaMA3-70B), and LangSmith for real-time tracing and debugging.
- Integrated YouTube and web content loaders (YoutubeLoader, UnstructuredURLLoader) with dynamic prompt templates to generate less than 300-word summaries using the summarization chain.
- Implemented secure .env-based configuration for Groq and LangSmith API keys, with support for runtime input, error handling, and modular loader/model customization.