Harshal Shrimali

LinkedIn/in/harshalshrimali Github Medium

shrimaliharshal@gmail.com

+1(408)-210-4370

Education

- **San Jose State University (SJSU)** — San Jose, CA
  Master of Science - Data Science — *August 2023 - Present*
- **Manipal University** — Jaipur, India
  Bachelor of Engineering - Information Technology — *August 2019 - May 2023*

Skills Summary

- **Programming Languages:** Python(Numpy, Pandas, scikit-learn), C++, Java, SQL, R
- **Databases & Cloud:** MongoDB, PostgreSQL, Snowflake, Cassandra, MySQL, Neo4j, AWS (Lambda, Sagemaker, S3 Bucket, EC2), GCP (Compute Engine, Buckets, BigQuery etc.), Azure
- **Big Data and Data Engineering:** Apache Airflow, Data Pipelines, Apache Spark, Pyspark, Hadoop, Hive, Flink, Kafka
- **Machine Learning and Deep Learning:** Keras, PyTorch, TensorFlow, NLP, NLTK, Spacy, XGBoost, LightGBM, LSTM, CNN, OpenCV, GANs, Transformers/LLM's - BERT, GPT, RoBERTa(Meta)
- **Tools & Frameworks:** PyMongo, Jupyter Notebook, MongoEngine, Postman, Git, GitLab, Jira, Flask, REST APIs, Microsoft Excel, Data Mining, ETL(Extract, Transform, Load), Streamlit, Kubernetes, CI/CD pipelines

Experience

- **Manipal University**
  Undergraduate Data Science Intern — *August 2021 - June 2023*

  Code Comprehension Using Langchain and LLM (Link)
  - Technologies: RAG,Generative AI, Python, LangChain, ChromaDB, LLM - OpenAI, Gemini
  - Built an automated **ETL pipeline** to ingest and process code repositories from GitHub, transforming code files into structured chunks for efficient querying and generation.
  - Created a **scalable vector database** using ChromaDB, enabling high-performance similarity searches and fast query retrieval.
  - Developed a conversational retrieval system using **LangChain and GPT-3.5-turbo** to enable interactive QA on code repositories, reducing manual documentation by over **50%**.

  Tradewise (Link)
  - Technologies: Python, Kafka, Docker, GCP, Firebase, Pyspark, Streamlit
  - Designed and implemented a **real-time data ingestion pipeline** using Apache Kafka to stream live stock data from external APIs(Alpaca and yfinance), ensuring reliable and low-latency data flow across system components
  - Developed scalable **ETL processes with PySpark** and MLlib to process over 1 million daily data points, transforming raw stock data into structured formats for analysis and machine learning.
  - Built a robust, containerized architecture using **Docker** and Docker Compose, ensuring seamless deployment and integration across cloud platforms on GCP and Firebase.

  BiasBeacon (Link)
  - Technologies: Python, Pytorch, Transfer Learning, Huggingface, AWS, Transformers - BERT, RoBERTa, LLama2
  - Implemented an automated data pipeline using Python and AWS to scrape and preprocess large-scale text data from Reddit, enabling efficient collection and storage for bias detection.
  - Fine-tuned large language models (LLMs) such as BERT, RoBERTa, and LLama2 using PyTorch, improving bias detection accuracy in political sentiment analysis to **82%**.
  - Employed mixed precision and differential learning rates during the fine-tuning process, achieving balanced model performance and reducing overfitting, reducing computational load and training time by **40%**.

  F1 insights(Link)
  - Technologies: Python, SQL, Pandas, Streamlit, PowerBI, FastF1 API
  - Designed and implemented an automated data ingestion pipeline using Python and FastF1 API, streamlining the collection and processing of real-time and historical race data. Utilized Pandas for data manipulation, enabling dynamic analysis of driver telemetry and team performance metrics.
  - Developed a Flask-based Formula 1 analysis dashboard, integrating FastF1 API and SQLAlchemy to process and visualize race data. Optimized SQL queries and implemented caching, resulting in a 35% improvement in data retrieval efficiency and enhanced user experience.
  - Created interactive visualizations using Matplotlib, Seaborn, and Plotly to display complex race metrics

Projects

- **Valorant Discord Bot**(Link)

  - Designed and developed a **data warehouse solution using BeautifulSoup** for web scraping esports data from vlr.gg, with **Apache Airflow** orchestrating scheduled data scraping and incremental loading, improving data update speeds by **2x**.

  - Built and optimized MongoDB schemas to efficiently store and retrieve tournament, player, and map-specific data, reducing query response times by **40%**.

  -Developed and deployed a Discord bot for esports data analysis, leveraging Pandas and MongoDB for efficient player and team comparisons, resulting in a **30%** increase in community engagement and caching mechanism to improve response time.

Publication and Certification

- **IEEE Publication - Context Based Recommender System**
- **Certificate - IBM Professional Data Science Course**
- **Certificate - IBM Advance Statistics**