

Kaspian Thomas

(985) 772-1721 | kaspian2005@gmail.com | [LinkedIn](#) | [GitHub](#)

EDUCATION

University of New Orleans | New Orleans, LA

May 2027

B.S. in Computer Science, Concentration in Machine Learning & AI | Cumulative GPA: 3.6/4.0 | CSCI GPA: 3.8/4.0

- **Relevant Courses:** Data Structures, Machine Structure & Assembly Language, Analysis of Algorithms, Natural Language Processing, Computer Networks & Telecommunications, Data Models & Database Systems, Python for Data Science & AI
- **Honors:** President's List x3 (Spring 2025, Fall 2025, Spring 2026), Dean's List (Fall 2024)

TECHNICAL SKILLS

Languages: Python, Java, C++, SQL, PL/SQL, JavaScript

Frameworks & Libraries: React, Node.js, TensorFlow, PyTorch, HuggingFace Transformers, Pandas, NumPy, XGBoost, imbalanced-learn, scikit-learn, spaCy, Swing, FastAPI, Pydantic, Uvicorn, Tailwind CSS, Vite

Tools: Docker, Git, PostgreSQL, SQLite, Oracle SQL, Jupyter/JupyterLab, Google Colab, LaTeX, Matplotlib, Langflow, Ollama, FastF1, Netlify, ngrok

Methodologies: Agile, Test-Driven Development, CI/CD, Client-Server Architecture, Algorithm Analysis

RELEVANT EXPERIENCE

F1 Real-Time Telemetry Commentary System | IBM Hackathon

May 2026

Co-authored · Python · FastAPI · WebSocket · React · IBM Granite · Langflow · SQLite · Netlify · [GitHub Repo](#)

- **Co-engineered** a gated RAG pipeline that distills ~1,400 lap events per race into ~64 grounded AI commentary moments (~4.5%), using IBM Granite (Ollama/Langflow) enriched with 3 years of official FastF1 data (64k+ laps) — **reducing LLM overhead by 95% with zero hallucination on race statistics**
- Engineered a real-time async WebSocket server (FastAPI) broadcasting **live telemetry, standings, and AI commentary** to all clients **simultaneously**; commentary generates as a fire-and-forget task so the 1-3s LLM call **never stalls the data stream**
- **Deployed end-to-end: React dashboard** live on Netlify, **FastAPI backend** exposed via ngrok static tunnel; architecture migrates from local IBM Granite to IBM watsonx.ai cloud with zero pipeline changes

PROJECTS

BioBERT Soccer Injury Severity Classifier

Spring 2026

Python · PyTorch · HuggingFace Transformers · scikit-learn · Matplotlib/Seaborn · Google Colab (T4 GPU) · LaTeX

- **Fine-tuned** BioBERT to **classify soccer injury reports** into 5 severity tiers using PyTorch and HuggingFace Transformers, training on Google Colab with a T4 GPU
- **Built** a full evaluation **pipeline** with **ROC/AUC metrics, confusion matrices, and precision-recall curves** via scikit-learn, matplotlib, and seaborn; compared against random and majority-class baselines
- **Authored** a full academic report in LaTeX and **analyzed/discussed model performance** with associates via presentation.

Customer Churn Prediction | CSCI 3220 Final Project

Summer 2026

Python · Pandas · NumPy · scikit-learn · XGBoost · imbalanced-learn · Matplotlib/Seaborn · Jupyter · Git

- **Built** an end-to-end ML **pipeline predicting customer churn** (7,043 customers) — EDA, feature engineering, class-imbalance handling, and model selection
- Compared **4 models** via **5-fold stratified CV**
- Handled 26.5% class imbalance with class weighting / scale_pos_weight, with **per-fold preprocessing to avoid SMOTE leakage**
- **Identified top churn drivers** (month-to-month contracts, fiber service, electronic-check payments); delivered a reproducible Git repo with README, report, and evaluation figures

Multiplayer Networked UNO Card Game

Spring 2026

Java · TCP/IP Socket Programming · Multi-threading · ExecutorService · Swing GUI

- **Implemented** a **multiplayer** networked UNO game in Java enforcing full UNO rules (special cards, draw logic, turn management) over **TCP/IP sockets**
- Designed a **multi-threaded server** using ExecutorService and ConcurrentHashMap with automated matchmaking, game room management, and countdown timers
- Built a **Swing GUI client** with card hand rendering, game log, draw mechanics, and an **integrated real-time chat panel**

Chatty – Multi-Client Chat Application

Spring 2026

Java · TCP Sockets · Multi-threading · Swing GUI · ConcurrentHashMap

- Developed a **real-time multi-client chat server** in Java handling concurrent connections via dedicated ClientHandler threads broadcasting to all users over **TCP**
- Enforced **unique** username **registration**, a live user roster, and graceful disconnect using a **thread-safe** ConcurrentHashMap; built a themed Swing GUI with EDT safety via SwingUtilities