AARYAN MEHTA

New York, NY | adm8315@nyu.edu | (646) 920-6974 | LinkedIn | GitHub

EDUCATION

New York University | New York, NY September 2023 – May 2025
Master of Computer Science GPA: 3.4/4.0

Mumbai University | Mumbai, India August 2019 – May 2023
Bachelor of Computer Engineering GPA: 8.7/10

TECHNICAL SKILLS

- ***Programming Languages:*** Python, Java, JavaScript, C, C++, R, Dart.
- ***Other Technologies:*** Machine Learning, Neural Networks, Generative AI, Large Language Models, AI Financial Modelling, Natural Language Processing, Computer Vision, TensorFlow, PyTorch, Keras, Scikit-learn, Numpy, OpenCV, Pandas, Matplotlib, Flask, Django, SQL, AWS, Docker, GitHub, Git, Jupyter, React, HTML, CSS, Firebase, Unreal Engine, Unity, Flutter, Adobe Photoshop.

PROFESSIONAL EXPERIENCE

Emkay Global Financial Services Ltd. Jan 2022 – May 2023
Machine Learning and SWE Intern *Mumbai, India*

- Deployed a predictive model utilizing LLMs (Large Language Models) and LSTMs (Long Short-Term Memory), that improved Emkay's equity trading strategies by 30%.
- This involved deep time series analysis of NASDAQ closing auction data and predicting the closing price movements of stocks using data from the order book and the closing auction of the stock.
- Integrated a Twitter sentiment analysis component using LLMs and NLP, which enhanced predictive accuracy by incorporating real-time market sentiment. This approach analyzed real-time stock-related news, leading to a further 15% increase in performance.
- Implemented efficient data preprocessing and memory optimization techniques, reducing dataset memory usage by 60% for over 200 stocks across multiple time intervals of the daily ten-minute closing auction on the NASDAQ stock exchange.
- Collaborated with the IT team to deploy several different models using Docker and AWS, ensuring seamless integration.

PROJECTS

GPT-2 Project: Optimized Model for Advanced Natural Language Processing
Tech Used: PyTorch, FineWeb-Edu Dataset, Multi-Headed Self-Attention Mechanisms, Dynamic Batching, Custom Logging

- Designed and built an optimized GPT-2 (124M) model from scratch, meticulously following the architecture guidelines from OpenAI's GPT-2 and GPT-3 papers.
- The network consisted of transformer blocks with multi-headed self-attention mechanisms, capturing complex text data patterns.
- Tokenized large text volumes using a custom tokenizer, reducing memory overhead significantly and setting up for robust training.
- Implemented multiprocessing for tokenization and dynamic learning rate adjustments, enhancing training efficiency and processing.
- Pretrained using hyperparameters from the GPT-3 study, focusing on learning rates and optimizer to maximize generation quality.
- Conducted extensive testing and tuning, evaluating the model's text generation against linguistic and contextual criteria, demonstrating improved performance and capability in natural language processing.

Foreign Whispers: YouTube Video Translation Platform
Tech Used: Python, Streamlit, OpenAI Whisper, MarianMT, PyTorch, TensorFlow, YouTube API, Pysrt, TTS, Moviepy.

- Engineered a website to translate YouTube videos from English to French, Utilized Python, Whisper for speech recognition, and MarianMT for translation, within a Streamlit-based UI.
- Developed a robust API to download YouTube videos and extract high-quality audio.
- Implemented OpenAI's Whisper model to convert audio to text, achieving above 95% accuracy in speech recognition.
- Integrated a neural machine translation system using MarianMT to translate English transcripts into French, that maintains 90% of the linguistic accuracy and context relevance.
- Created a dynamic text-to-speech module to convert translated text to audio, syncing it perfectly with the original video timeline.
- Designed and launched a user-friendly web interface using Streamlit, which allows users to input YouTube URLs and view accurate translations for each timestamp.

Deep Reinforcement Learning for Stock Trading from Scratch
Tech Used: Python, FinRL, OpenAI Gym, Stable Baselines, Yahoo Finance API, Pandas, NumPy, Matplotlib.

- Developed an automated trading solution using Deep Reinforcement Learning (DRL) to maximize portfolio values in a simulated stock trading environment, framed as a Markov Decision Process (MDP).
- Constructed and trained using DRL algorithms, achieving dynamic stock trading strategies by evaluating actions (buy, hold, sell) across multiple shares within the Dow 30 constituents.
- Utilized the FinRL library for building a robust training pipeline with Yahoo Finance API to preprocess data with custom indicators like MACD and RSI.
- Integrated machine learning frameworks like OpenAI Gym for environment simulation and Stable Baselines for implementing multiple DRL algorithms.
- Applied ensemble strategies to optimize trading actions based on rolling-window performance, increasing the model accuracy.
- Conducted rigorous backtesting using the Quantopian Pyfolio package, demonstrating the model's effectiveness through enhanced financial metrics and robustness against market variations.

Credit Card Default Prediction for American Express Customers

Tech Used: Python, TensorFlow, Keras, Scikit-Learn, Pandas, Matplotlib, Seaborn, NumPy, Gated Recurrent Neural Networks.

- Led the development of a predictive model using Gated Recurrent Units to predict the probability that a customer does not pay back their credit card balance amount in the future based on their monthly customer profile.
- Achieved top quartile performance on hidden American Express test data, surpassing baseline models.
- Implemented a custom loss function for class imbalance, improving model sensitivity towards minority classes by 20% when compared to standard loss functions.
- Engineered the most relevant features from a dataset containing 6.5 million records of time-series behavioral data and anonymized customer profile information of American Express customers by reducing data dimensionality which enhanced model performance.
- Automated the data preprocessing pipeline to handle multiple CSV files, resulting in a 30% reduction in preprocessing time.

AI-Based Skin Lesion Analysis for Early Cancer Detection

Tech Used: Python, LightGBM, Scikit-Learn, Pandas, NumPy, Matplotlib, OriginalEncoder.

- Developed a machine learning model to differentiate between malignant and benign skin lesions using image data resembling smartphone photos, as part of a Kaggle competition aimed at improving cancer diagnostics in telehealth settings.
- Engineered robust feature extraction methods to enhance model accuracy on low-quality images, including ratios of lesion dimensions, contrast metrics, and 3D spatial positioning, improving the diagnostic precision by 10-25%.
- Optimized model parameters to focus on maximizing the partial area under the ROC curve above an 80% true positive rate, crucial for clinical relevance. This approach improved the detection sensitivity of early-stage malignant lesions by over 20%.
- Achieved a top 5% ranking in Kaggle competition, demonstrating the model’s effectiveness across a novel dataset that spans lesions from thousands of patients across three continents, showcasing robustness and scalability.
- Utilized Python, LightGBM, and scikit-learn for data preprocessing, model training, and evaluation.

“Monet”-ize Your Photos using GANs

Tech Used: Generative Adversarial Networks(GANs), CycleGAN, Python, TensorFlow, Keras, PyTorch, Matplotlib, Numpy.

- Developed an artistic AI that learns to paint like Claude Monet, transforming photos into impressionist art, making art accessible and fun by blending classic styles with modern images.
- Achieved a style transfer accuracy of 90% as measured by aesthetic and stylistic evaluations.
- Developed a CycleGAN model to transform 1000+ realistic photographs into Monet-style paintings.
- Engineered the model using TPU strategies to enhance computational efficiency, achieving 15-20% performance improvements.
- Implemented custom data loading and preprocessing functions to handle TFRecord datasets, resulting in a 30% improvement in data throughput during training sessions.
- Utilized advanced neural network architectures like U-Net and instance normalization for the generator and discriminator models, optimizing the fidelity of style transfer.
- Extensive testing and refinement of model parameters to minimize cycle-consistency loss, enhancing model performance.
- Visualized transformation results to demonstrate the model's effectiveness in replicating Claude Monet's iconic style.