Avani **Sharma**

☎ (385)-216-2582 | ✉ avaninith@gmail.com | Avani1994 | @avaninith | sharmaavani

Education

University of Utah
GPA - 3.98

MS in Computer Science (Specialization: Machine Learning) — *Aug. 2016 - May 2018*

- Masters Thesis : Topological Analysis and Visualization of Mice Temperature Data for Exploring Biological Events
- Relevant Course Work: Machine Learning, Statistics, Deep Learning, Natural Language Processing, Data Mining
- Coursera Deep Learning Specialization: Improving Deep Neural Nets - Hyperparameter Tuning, Sequence Models

NITH (National Institute of Technology, Hamirpur)
GPA - 9.18

B.Tech. in Computer Science and Engineering — *Aug. 2012 - May. 2016*

Skills

Programming	**Python, Javascript, C++, React, R, SQL**
Libraries	**Numpy, Pandas, Spacy, Scikit Learn, MLflow, Pytorch, MLOps HuggingFace, Keras, NLTK, matplotlib, seaborn, opencv, PIL, BeautifulSoup, Textblob, TAlib**

Work Experience

Bloomberg LP
Remote

Software Engineer (ML + Backend) — *Feb. 2021 - Present*

- Contributed majorly to Bond Pricing and Spread Analytics Team. Coding in **Python, C++**
- Developed ML Models to predict Spread of Bonds using Market Data and Price making pricing **80% faster**.
- Experimented with several ml models including perceptron, random forests, linear regression, with regularization, deep neural nets and evaluated results using RMS loss. Linear Regression using polynomial coefficients performed best with RMS error less then 10bps. Models were trained separately for each of the 25+ ETFs. (**Pytorch, Numpy, Pandas, Scikit Learn, matplotlib**)
- Built infrastructure to support storing serialized **ML ETF pricer model** in DB and loading them 50% faster for pricing (C++ and Python).
- Automated ML model training pipeline to allow training for a set of dates for all ETF cusips (**Python, scikit-learn**) reducing 100% manual labor.
- Created Jupyter notebooks for interactive debugging increasing sprint efficiency by 5 points.
- Implemented Bond structuring and Market Keys Structuring as a microservice to support the pricing of floating Bonds. (**C++**, **integration test in python**)

Messagink
Remote

Machine Learning Engineer — *Jun. 2020 - July. 2021*

- Built a feature of importing textual conversational story from Whatsapp / Chat Images to textual conversation. This project was divided into three parts where first part was extracting text from chat images and showing them in a nice chat format on the website. We used **Tesseract OCR** and basic maths to solve this problem.
- Second part was detecting emojis in the chat images. Experimented with different approaches including CNNs and classical image processing. Classic image processing yielded best results. We used **SIFT features for matching** . We also integrated this model with the text recognition model in first part.
- Last part was noise removal and dark mode integration. Used **regex** for time stamp removal, patching (black / white) to remove unwanted noise in place of emojis in tesseract OCR and masking dark images, as Tesseract OCR works fine with black text over white background

Goldman Sachs
Salt Lake City, UT

Computer Scientist (Machine Learning) — *Jul. 2018 - Jan. 2021*

- Developed Stock Selection Framework using Technical Indicators. Curated dataset with the help of basic stock metrics **(OHLCV)** and **TALIB**. Trained ML models (SVM, Multi Layered Perceptron, Random Forest) on the curated data to learn whether trader should be buying the stock if gain after 3 days of buying is > 2.5%. Also backtested the model output on test data to compare the loss-gain using different models. **Multi Layer Perceptron** performed best on all the stock tickers we had. In most of the cases it stayed higher than buy and hold strategy.
- Deployed Stock Selection framework using **MLFlow** for automated fetching of data, preprocessing, training and inference. This allowed **reproducibility, versioning and comparison** with previous models.
- Experimented with different approaches for building an **Anomaly Detector** to flag outliers and clean the trading data, allowing business teams to exclude/include data points in reports. Used BoxPlots, Quartiles, SD, Median, Elbow method for exploratory analyses. And **K-means (unsupervised), KNN (supervised)** and **Isolation Forests** formed the core of anomaly detector.

SCI Institute UofU
Salt Lake City, UT

Graduate Research Assistant — *Jan. 2017 - May. 2018*

- Worked on Classification of Autism funded by **NIH** and achieved accuracy of **71%** using **Scikit learn's** Support Vector Machine classifier on top of **Correlation and Topological** features obtained after preprocessing dataset using **TDA-R, Numpy, Pandas**

Projects

2023 **Reinforcement Learning for Stock Investment**, Researched and Built an RL Agent based on OHLCV Environment to guide on the amount that should be invested each day based on how market is doing

2023 **Fitnets Replication**, Implemented Fitnets in Pytorch for knowledge distillation after reading research papers

2023 **Response selection for Ubuntu Support System Chats**, Researched state of art techniques for Response selection (Semantic Search) in ChatBot development. Currently working on training a deep learning model in **pytorch** which will find most similar response from historical chats for a given user query. Dataset: Ubuntu Dialogue Corpus

2022 **Intent Recognition for Banking Support Chats**, Researched state of art techniques for ChatBot development. Trained an intent classification language model by fine tuning BERT model on banking dataset in **pytorch**

2022 **Multilabel classification using BERT**, Finetuned BERT Base Model for Toxic comment classification on Toxic Comment Dataset using **Pytorch Lightning**

2020 **AI in Drone** , A drone installed with voice and facial recognition to follow the specific person and listen to voice commands.

2017 **University Webpages Clustering and Visualization**, Employed tf-idf with clustering techniques such as **k-means, Agglomerative via Sklearn** to group University Webpages and outperformed **bag of words**, taking accuracy from **25% to 75%**

2017 **Splitting Convolution Networks and training them on Multiple GPUs**, reducing training time by **50% to 74%** using no communication/ hybrid communication scheme within tolerable accuracy drop of **2% to 12%**

2016 **Author Attribution using Multi Class classification (ML)**, Leveraged **Scikit Learn's** SVM 1-vs-1 and 1-vs-rest classifiers in **Python** over Amazon Commerce Reviews Dataset, with accuracy of **97% and 96%** respectively and visualized precision and recall using **matplotlib**

2016 **Coreference Resolution(NLP)**, Implemented modules: exact/partial/pronoun matching, semantic class match, appositives match using **NLTK, wordnet** and later clustering in **python** to achieve **61%** accuracy among **top 5** in the class of **120 students**

2016 **Web Page Recommendation System**, Implemented hybrid system using **content based filtering collaborative filtering**

Publication

Chat Images to Textual Conversation: Text Recognition *SLC, USA*
Chat Image Parsing to recognize texts and structure them in Text Message Format *June. 2020*

Emoji Detection and Recognition in Whatsapp Images *SLC, USA*
Detected Emoji Bounding Boxes, Assigned Labels and integrated them with text recognition *August. 2020*

International Journal of Advances in Electronics and Computer Science (IJAECS) *HP, India*
Improving Page Ranking For Search Engines by accounting for **link visibility and link position** in the algorithm. *May. 2016*