NISTHA MITRA

University of Maryland 2022

nmitra28@umd.edu | College Park, MD | nmitra28 | nmitra28

EDUCATION

University of Maryland: B.S. in Computer Science (GPA: 3.4/4.0)

Relevant Coursework: Advance Algorithms, Object Oriented Programming, Advance Data Structures, Computer Networks, Data Science, Machine Learning, Computer System

EXPERIENCE

Data Science Intern

Oracle Corporation, Oracle Cloud

June 2021 – September 2021 | Virtual

- Used a 3D Convolutional Neural Network to predict lungs scarring on CT scan images of Covid Patients and a credit card fraud detection model using OCI AutoML.
  CT-Scan Prediction Notebook and **Fraud Detection Notebook**
- Oracle Cloud Services, Object Storage, AutoML and Accelerated Datascience SDK, Keras, TensorFlow, OCI Data Science Platform, Seaborn, Numpy
- Encapsulated the projects in a HTML/CSS website to create an end to end tutorial for OCI users and clients. **(web page)**
- Worked with an Oracle client to create a fault tolerant, secure Cloud architecture model to solve their security breach problem.

Undergraduate Research Assistant

UMD Computer Science Department, Data Visualization Research

Jan 2021 – November 2021 | College Park, MD

- Analysis of temporal event sequence data using different data visualization tools available and build a guideline to recommend the appropriate ones for different data sets.
- 6-12hrs/ week doing in-depth literature review, comparing visualizing tools and techniques, and analyzing their performance in real-world data sets.

Software Engineering Intern

Novuz Inc.

May 2020 – August 2020 | Virtual

- Developing a parsing tool used to make an automated machine to convert legacy applications to cloud-based apps
- Python, Lex and Yacc packages, Regex, Linux OS

Data Science Intern

TATA Steel, TATA Groups

May 2019 – August 2019 | Tatanagar (Jamshedpur), India

- Used supervised learning models using Python to analyze 98000 data points of employees to predict their attendance in various training programs offered by the company **(See Notebook)**
- Goal: Make efficient training programs and target them to specific departments
- Jupyter Notebook. Packages used: Pandas, Seaborn , matplotlib, Regression models, Decision Tree models.

SKILLS

- **Programming Languages:** Java, C/C++, Python, HTML, CSS, JavaScript, SQL, MATLAB, Dart
- **Front End:** Flutter, HTML5, CSS3, Figma, Sketch
- **Back End/Cloud:** MongoDB, Firebase, RESTful APIs, OCI, AWS, GCP
- **Technologies, OS, Libraries:** Wireshark, Protocol Buffers, VMWare, Git, NoSQL, MySQL, LaTeX, Tensorflow, Keras, Pandas, Numpy, Matplotlib

PROJECTS

- **Black Lives Matter : Fatal Police Shooting Analysis**
  A Data Analysis Pipeline Documentation to understand and follow the steps of Data Science: data scraping, cleaning/processing to different kinds of visualization and inference.
- **Moneyball Analysis (Oakland Athletics)**
  Analysis of how efficient teams have been historically spending money and getting wins in return. However, Oakland Athletics performed extremely well between 2000-2005. Using data I analyse and observe this anomaly and pattern.
- **BitTorrent Protocol**
  Implemented a BitTorrent (Peer-to-Peer file sharing protocol) client that facilitates faster downloads for larger and popular files using Python.
- **Chord Protocol**
  Implemented the Chord Protocol Algorithm for a Peer-to-Peer Distributed Hash Table that optimizes lookup time from O (N) to O (log N) and guarantees no single point of failure using C and Google Protocol Buffers.
- **Chat Server**
  Implemented a chat server that allows clients to engage in group chats in chat rooms and send private messages to one another. Used language C++ and UDP Protocol.
- **TCP Sever/Client**
  Designed a TCP-Server to encrypt various segments of the data inside the file using the AES and a TCP-Client to test.
- **UDP Sever/Client**
  UDP client and server to run a simplified version of NTP (Network Time Protocol).
- **Exposing Flaws in the Cloud**
  Used Google Spanner to test the efficiency in terms of latency and throughput by sending transactions from different geographical standpoints.