Prakhar Gandhi

Github | Linkedin | my site | Portfolio | gprakhar0@gmail.com | 9133234325

micro1 certified

Education

July 2015 - Dec 2019 B.Tech. (Bachelor's of Electronics and Instrumentation) from
Birla Institute of Technology and Science, Pilani, Hyderabad Campus, India (GPA: 7.15/10.0)

Coursework Data Structures and Algorithms, Neural Networks and Fuzzy Logic, Discrete Math for CS, Digital Image Processing
Self Taught Courses Database Management Systems, Number Theory: Project Euler
Office Courses Deep Learning: Advanced NLP and CNNs, RNNs, LSTMs and GRUs, Bayesian Machine Learning in Python: A/B Testing, Recommender Systems and Deep Learning in Python, CNN in python-Computer Vision Certificate Link
Gold Award Have received a gold award from Standard Chartered Bank Global Business Services Private Limited for a project done as a part of Hackathon

Technical Skills

- **Languages** Python, C++, JavaScript, SQL, CSS, HTML, Latex, Object Oriented Programming, Data Structures And Algorithms, Data analysis
- **Databases** PostgreSQL, MongoDB, Cloud Firestore, Firebase
- **Machine Learning** Scikit-learn, Pandas, Numpy, Keras, Tensorflow, Pytorch, SciPy, NLTK, matplotlib, seaborn, Probability and Statistics, decision trees, survival analysis, time series analysis, Neural networks
- **Tools && Frameworks** Flask, Django, pySpark, github, Jira, Dash, Bitbucket, Docker, Azure functions

Experience

Wipro Private Limited with Ericsson as client - Senior Software Engineer (Python Developer) Feb 2025 - Current

- **GenAI Product Creation (ENIQ AI UI) | Python, dash, data structure, pandas, scikit-learn, ruff**
  - Built **ChatGPT-style user interface** with **CRUD features** in **Dash + SQLite**.
- Demo presented to **1000+ Ericsson employees**.
- Quality Features included **Pagination, Dark/Light Theme Toggle, Export to CSV support for tables, Interactive Zoomable/Drillable Graphs (Plotly), Textual/Tabular/Graph formatting**.
- Learned skills like **LLM-Based Evaluation, RAG, pydantic, vector databases, chunking and retrieval, Reinforcement Learning Human Feedback, recursive text splitter, langchain, openai-models used GPT-4/3.5.**

Wipro Private Limited with Mercedes Benz as client - Senior Software Engineer (Python Developer) May 2024 - Jan 2025

- **Python Scripting (Mercedes Benz Products) | Python, mpire, data structure, tree-sitter-cpp, tree-sitter, subprocess, shutil, pandas, clang-tidy, shellcheck, ruff**
  - managed automation of manual cpp file code generation by doing **Parameter and function name extraction of car product cpp files** using **python and multiprocessing**. This work determined whether the function signature is read or write using **regex**. Further Created a cpp code parser in python to extract function name and first parameter as a key value pair. After that manipulated the cpp code read and write and got the list of parameters of direction out.
- Created a **flask app** that includes **live terminal** using **socketio** and **docker** integration in end to end **GUI** from the team.

Standard Chartered Bank GBS - Senior Data Analyst (Data Analyst II),Associate Projects (Data Science) Aug 2020 - Jan 2024

- **Python Scripting (Banking) | Python, pandas, numpy, pdoc, opencv**
  - Have created a **scalable microservice** using parallel processing by storing data in a pandas dataframe. The logic developed was as follows:-
* Stored list of text coming from POST request in a column in pandas dataframe. * Set the number of workers considering default value into account. * Created a function that takes a text and generate features. * Called this function using apply of pandarallel package. * Extracted output in a list of dictionary format once parallel processing is done. * Verified the output using postman.
  - Have created a **QA generation pipeline** from scratch using the knowledge of competitive programming. The logic developed was as follows:-
* Extracted text from pdf first rowwise then columnwise sorted. * Converted context into mini paragraphs by the logic of CP Algorithms. * Applied parallel processing to the list of mini paragraphs by invoking the **Large Language Models** named **t5-base-squad** of **medium parameter size** only once by keeping it in **global scope**. * If size of list of contexts was less than 2 applied normal processing instead of **parallel processing**. * The context size was chosen such that random lookup made sense to the LLMs with high enough accuracy. * Then returned the consolidated output to the excel file as a part of automation.
  - Automated synthetic dataset generation by doing **Multi blending labels of product documents**(watermarks, barcodes, seals and other confidential details) using **python and opencv**. This work involved Parallel Processing improved Data Creation for the agreed deliverables from **4 hour** to **1 hour** per document 256 combinations of logos.
  - Implemented **multiprocessing** on django microservices by using parallel processing using **pandarallel** package and decreased the execution time.
  - Implemented **multiprocessing** at hackathon on Large Language models using transformers(t5-large model) to generate pairs of questions and answers on a set of company pdf files.
  - Implemented **McNemar Test via statistical analysis** for cross-team members from structured data on ML Pipeline to compare the performance of 2 classifiers and selecting the best model further integrating it within the champion challenger pipeline.
- **Ner Explainability Creation | Python, sklearn, pandas, numpy, plotly**
  - Researched on methods to detect explainability of an existing system at word level accuracy of 80%.
  - Implemented NLP pipeline for a ML based NER explainer search engine. Developed text preprocessing functions and integrated them in pipeline to prepare input for a machine learning model during training as well as prediction.
  - Deployed local level explainability on trained SKLEARN CRFSUITE model using dash app on localhost interface to handling and improving query classwise visualization response for a single sentence accurately.
  - Created Champion Challenger and added NER pipeline for existing developed pipelines by comparing champion log files and challenger log files for all of the best selected metrics for ML with regression and classification, Object detection with/without classification and Text Prediction models using python. Reduced overall time complexity to consider Million points(scalable). Further selected best metrics for Object detection pipelines.
  - Communicated and presented insights clearly and compellingly to senior leadership of the organization
- **Image duplicate prediction (Banking) | Python, pandas, numpy, tensorflow(imagededup), pytorch(detecto), opencv**
  - Developed analytical and ML models to assist other insider client team in identifying the right duplicates and influential missing documents for various banking pdfs to improve the overall duplicate detection of barcodes, seals and other important contents of the TradeLC Docs by atleast 80%.
  - Built ML models to get probability of bounding box detection of barcodes, watermark seals, handwritten signatures (implemented Structural Similarity Index
    (SSIM) score on top of cosine similarity score before comparison of two images) Projected the bounding box to dataset and retransformed the data to clients. Achieved 85% Accuracy score using transfer learning of faster rcnn mobilenet models on neural nets.
  - Developed data visualization, report creation of analytical models to support our hypothesis and for client presentation.
  - Selected best metrics for object detection with or without classification and deployed same into production in champion challenger pipeline.

Experience with Pyspark

- **Pyspark Zycus assignment**
  - Built an automated text classification pipeline to classify products on the basis of text reviews provided by the users. Github
  - Improved data processing speed by 91% by creating a parallel computing pipeline to convert a column of paragraphs into [tokenizer to stopwords remover to vectorizer to inverse document frequency to logistic regression model] and get the final predictions.
  - Filtered out the labels in code to remove **NaN labels** to improve the final evaluation metrics and finally applied the performance metrics like accuracy, weighted precision, weighted recall, area under curve and f1 score to obtain the final verdict.

Other Technical Skills

- **Other Skills includes which are but not limited to :-** •• AI/ML Innovation Meet Attender, RESTful APIs, NOSQL, SQL, Custom Chart Webapp, Backend, Debugging, Programming, Monitoring, Python Testing

Projects

- **Text Robustness app | Python, Flask, Spacy** Office private project
  - Developed a flask app to visualize model predictions and flipped classes based on synonym replacement, insertion and deletion of tense words of banking words present in a general spacy dictionary to check the robustness of model to any such attacks.
- **NER/POS Tagging Web Application | Python, Spacy, numpy, pandas, Flask, Dash, HTML/CSS, JS** Office Private Project
  - Implemented Spacy models to tag all words of paragraph of a text column in csv datasets to their Named Entity or Part of Speech.
  - Automated an web interface using Flask API to interact with the model and finally deployed app.
- **Image Global Explainability | Python,Networkx,Dash** Youtube Link to Demo
  - Developed an dash app for a similar class visualization of a particular image in networkx graph based on custom features to visualize the connections.
- **Automated Topic Modelling App using BERT | Python,BERTopic,Dash** Youtube Link to Demo
  - Developed an dash app for a topic modelling visualization of the text column in pandas dataframe in the context of unsupervised learning(can include supervised case as well).
- **Bayesian Optimization to increase the metrics of bad performant model | Python, Hyperopt, LazyPredict** Link to Demo
  - Developed best performant globally optimized models from the list of models whose accuracy is far lower from baseline set accuracy(can include regression case as well and other metrics like f1 score can be taken as well).
- **Python code generator using Large Language Model | Python, transformers, pytorch** Github Link to Demo
  - Implemented magicoder prompt for instruction in responses using generator with temperature, deployed at github account and visualized the various types of codes and finally reduced the uploading time by making it work only for smaller max length wherever needed of original prompt. In the end filtered out only the code snippet from the output.
  - Same idea can be scaled to detect other known types of prompts by using parallel processing and different temperature values.
- **Avengers Face Detector Deployed App | Python, Sklearn, dlib, face-recognition** Link to Demo
  - Implemented avengers face detection app in Python using machine learning with encoded feature vector generation of training dataset, deployed at huggingface and reduced the response time to 10 ms and finally achieved 95% accuracy.
  - Same idea can be scaled to detect a thousands of faces by optimizing the code using numpy vectorization techniques.

Achievements

- red coder on atcoder. Link

- max 3* rating on codechef, Link.
- max 1368 rating on codeforces, Link.
- created a 3 layer artificial neural network from scratch as a part of research paper provided to me in coding assignment from the knowledge gained from udemy courses and gained 96% accuracy for benchmarks like iris dataset.(code can be run for mnist dataset as well), Link.
- converted Matlab to Python code for all of the ML algorithms from scratch as a part of BITS Course NNFL in various coding assignments from the knowledge gained from udemy courses and gained 81%-96% accuracy for assigned datasets.(included stacked autoencoders with backprop), Link.
- created chatbots in python that answers the query of integral calculus and differential calculus in form of equations, Link.

- Guided a senior colleague in visualizing robustness check of ML/DL model at a targetted classwise image level attack and made a opacity test to compare a permutation of two classes at a time by implementing the same from the paper. Further Detected poison in image dataset. Paper Link.

Small Startup internship July 2018 - Dec 2018

- **Backend Development/Python Development/Software Development/Web Scraping/Documentation generation**
  - Built an automated monitoring system to transfer one project to another over time on NoSQL cloud firestore db.Github
  - Managed**webscraping** to scrape kirana datasets of sites like justdial.com over multiple metric definitions and org-ids to increase scalability of our own product by 150 times.Extracted the hidden elements of phone numbers of dynamically loading scripted websites like justdial with lxml parsing.
  - Automated **sessions, cookies, and web authentication** mechanisms via python code while logging into mobile friendly version of facebook website.
  - Performed and Simulated user interactions on the web pages of facebook.
  - Performed Google search Automation to find missing product info.
  - Navigated through Firewalls, User Logins, and requirements of facebook mbasic website using random time sleep function to fool the website into thinking there is no web-crawler crawling the website.
  - Developed code to extract all the friends of user on facebook and learnt the use of **recursion** to support daily runs on production of barcodes by reducing the time complexity using memoization technique and created **tkinter responsive app** for the same that uses the concept of **multithreading**.

Give an ideal to strive towards sun...(GREATEST PYTHON PACKAGES) inspired by openai team respectively sam altman

• AUTOMATE SCALABLE UNSUPERVISED DATASET GENERATIONLink • AUTOMATE UNSUPERVISED DATASET GENERATION Link