Harshit Sethi

hstsethi@outlook.com | linkedin.com/in/hstsethi | hstsethi.vercel.app | github.com/hstsethi | kaggle.com/hstsethi

Education

Vivekananda Global University Jaipur, Rajasthan, India
Bachelor of Computer Applications in Artificial Intelligence, 7.6/10 *2024-07-XX - 2028-05-XX*

Technical Skills

Languages: C++, Python, Bash
Developer Tools: Git, NVim, Jupyter
Skills: SEO, Equity Investing(Cap 1.10k+ INR at 18), memoir), Linux Command Line, Regular Expressions, Leadership Skills(Found 8+ tech, educational groups, memoir)
Libraries: Pandas, Numpy, Matplotlib, Scikit-learn

Experience

Models Limited Public Beta Tester 2024-09-XX - 2024-10-XX
Github

- Reported a bug about different models being served through API as compared to of Playground. Collected output of various models including Mistral-large, gpt4-0, Phi-3.5-instruct-mini by using Azure Inference SDK through Python to bulk prompt models and compared their output to output of playground.
- Improved multilingual model search process by identifying and reporting lack of multilingual tags in Mistral 2407. Supported claim with official model card, MMLU benchmarks, and personal testing in Hindi, Punjabi with literary works.

Open Source Contributor 2021-XX-XX - Present
Open Source Community

- **Free Media Heck Yeah:** Made very first contributions, pull request number 9, 10 in 2021, aged 15.
- **Hacktoberfest 2024:** Completed challenge with 6 merged pull requests, including **4 in top 5 highest stared projects**. Writeup
- **Telegram Leecher:** Improved download speed of aria2c by **15%** by various hacks, sourced from personal dotfiles repository, including decreasing split size, increasing maximum connection per server limit.
- **Weblate:** Actively translating various open source projects into English, Hindi, Punjabi

Projects

DietCli | *C++, Cmake, RapidJSON, Python, Pandas, Matplotlib* 2023-03-XX

- Developed a cross-platform, command-line nutrition tracker with C++ using RapidJSON as a JSON parser, std::ostringstream for buffers, Cmake as build system, USDA RDI as a data source. It completes the full run on non-minifed JSON, including calculations, printing, not just parsing, in **001s**. Kaggle Benchmark
- Increased Input/Output speed by **35%** by utilizing const references, RapidJSON and buffering via std::ostringstream. Instead of unbuffered std::cout and INIParser used in NutritionCLI.
- Created a Python program with Pandas, Matplotlib to calculate and generate data visualizations.

Awesome J2ME | *Java, J2ME, Github Actions* 2023-07-XX

- Released a curated list of **50+ items**, related to J2ME. Employed Awesome Lint with Github actions. Single-handedly refactored it again within 7 days, reviewed 3 other pull requests. Passed **all 33 automated tests** in first try, which has rate of only 54%Source, writeup. Got merged into main Awesome repository which has 400k+ stars. Featured into Hackclub. Gained **115+ stars**, 5+ contributors.
- Created a similar list, Awesome Symbian, which gained **20 stars** in one month.

Personal Website | *HTML, CSS, Bash, Python* 2024-02-XX

- Authored **10+ unique articles** on various topics including career, finance, lifestyle, autobiography, programming, philosophy employing empirical, rational evidence, charts and persuasive writing.
- Improved SERP ranking across all major search engines including Google, Bing, Yandex, Duck Duck Go, Brave Search, by keyword research, meta tags, schema data, collaborating with webmasters of minimalist web directories, promoting it on social media platforms, own groups.

in-mob-prefix | *Python, Jupyter, Pandas, Matplotlib, Bash, Scikit-learn, Numpy, Joblib, Argparse* 2024-09-XX

- Created 4 CSV dataset, charts, machine learning based Python program about mobile number prefixes in India, that got featured in Awesome OSINT(20,000+ stars). Utilized Jupyter, Matplotlib for visualizations. Scikit-learn, Pandas, Numpy for machine learning. Preprocessed raw data using various techniques including Pandas, Microsoft Excel, SQL, Regular Expressions.

- Achieved **15%** increase in accuracy by using a combination of dataset lookup, and training 4 specialized gradient boosting classifiers models on 4 different set of prefixes.
- Achieved **20%** increase in performance speed by chaining Pandas queries, efficient use of Numpy array, creating and loading models from disk using Joblib.