Nakul Gangolli

ngangolli108@gmail.com | 201-247-4231 | linkedin.com/in/nakul-gangolli| US Citizenship

Summary of Skills:

Coding: Python (Numpy, Scipy, Pandas, SciKit-Learn, SciKit-Image, OpenCV, Matplotlib, Seaborn, Multiprocessing, TensorFlow, PyTorch, Keras), Bash, SQL, C/C++, Google Colab, Jupyter Notebook

Computing on Cluster/Cloud Environments: Pipeline development, maintenance, management, and orchestration, Knowledge of computing with AWS, Azure, and GCP, and using Databricks

Statistical Analysis and Machine Learning: Bayesian Inference, Clustering Analysis, Regression, Hypotheses Testing, A/B Testing, Data Visualization, Natural Language Processing, Large Language Models

PROFESSIONAL EXPERIENCE

Department of Physics, UC Riverside

Implementing and Optimizing Semi-analytic Models at Scale February 2024-Current

- Led a cross-functional team of astronomers and astrophysicists to implement large-scale simulations, analyzing galaxy formation and evolution and their relation to astronomical observables
- Created multiple large-scale simulations to generate critical datasets, totaling to ~500TB, delivering actionable insights to stakeholders on how varying model parameters, influence key astrophysical metrics, optimizing future analysis strategies
- Applied time-series analysis to model stochastic star formation rates datasets, enhancing output accuracy and increasing detection of starbursts by approximately 10%, displaying data-driven decision making and predictive insights
- Increased the speed of Python/C++ scripts by factors scaleable to the number of cores and streamlined C++ pipelines using wrappers in Python to populate halos and galaxies below the resolution limit of simulations
- Updated and optimized makefile options to increase production runs of large simulations by ~10-15%
- Debugged simulation codes and implemented tests that reduced run times by factor of 10, allowing for more robust and expansive parameter space exploration

Detecting Correlations Between Independent Observables June 2023-August 2024

- Implemented regression and clustering analysis to correlate the environments of galaxies to their observed properties, and extended current observational constraints to limit models
- Analyzed data patterns to identify how external factors impact galaxy characteristics, leading to improved predictive models and stronger validation methods
- Created data pipelines, using Python and Bash and orchestrated with SLURM, to preselect simulated galaxies and run models based on observational datasets and reduced time to create thousands of catalogs by 75%
- Implemented hypothesis and A/B testing using a Bayesian framework to examine if our improved modeling required modifications, concluding our models are adequate to explain rarified observations (to within 10%)
- Created data visualizations using Matplotlib and Seaborn and identified how the correlation between distinct, independent observables were dependent on local environmental properties
- Developed data visualizations using Matplotlib and Seaborn to analyze correlations between independent observables, highlighting their dependence on local environmental factors.
- Maintained, created, and structures pipelines to implement multiple models effectively over multi-core and multi-node environments
- Designed, built, and maintained pipelines to efficiently execute multiple statistical models across multi-core and multi-node environments, optimizing performance and scalability.
- Modified and implemented pipelines to reduce runtimes of future analysis by 50%, and modified outputs to reduce storage space by ~40%

Data Pipeline Maintenance and Feature Extraction from Simulations August 2021-Current

- Managed data pipelines for efficient storage, transfer, extraction, cleaning, transformation, and loading (ETL processes) for simulated, petabyte-scale data sets
- Extracted, loaded and transformed data from petabyte-scale data sets and implemented feature engineering to find correlations between different galaxy properties

- Extracted relevant data from petabyte-scale datasets and reduced storage space/transfer times by 90% through pre-selection techniques
- Stored, managed, and warehoused ~500TB of data, distributed on multiple NSF supercomputing clusters using Bash orchestrated with SLURM
- Used Bash scripts to optimize and automate packing and transfer routines distributed over multiple cores, freeing up ~3 months of wall clock time
- Restructured previous pipelines and files by reformatting and batching of data, reducing wallclock time by ~50% and reduced RAM usage by ~20%
- Developed and optimized parallelized Bash/Python scripts, orchestrated with SLURM, to automate analysis pipelines and reduce run times based on available core capacity
- Enhanced existing pipelines by introducing customizable headers, reducing the need for code modifications and enabling seamless use across the team with minimal training.
- Increased the speed of in Python/C++ scripts by factors scaleable to the number of cores and streamlined pipelines using Python wrappers to populate halos and galaxies below the resolution limit of simulations
- Increased the resolution in post-processing of simulations by x10, by using models implementing Bayesian framework for density fields using C++ and Python
- Increased the speed of in Python/C++ scripts by factors scaleable to the number of cores and streamlined pipelines using Python wrappers to populate halos and galaxies below the resolution limit of simulations
- Improved previous algorithms to create data below resolution limits by implementing multiprocessing and multithreading techniques

Clustering Analysis of Galaxies for Void/Cluster Detection January 2019 - November 2021

- Queried and cleaned cosmological simulations from various online sources using SQL (PostgreSQL)
- Classified and detected clusters/voids in simulated fields by using image detection (image segmentation, peak finding, and Voronoi tessellations) and clustering (K-Means Clustering and DBSCAN) techniques using SciKit-Image, OpenCV, and SciKit-Learn
- Developed statistical metrics and implemented hypothesis testing, using Bayesian inferences, to inform and optimize future observational strategies aimed at distinguishing between models
- Tested the viability different statistical metrics (such as number density and clustering) to distinguish between different models of galaxy formation and evolution
- Enhanced clustering algorithm efficiency by 80%, enabling the processing of hundreds of field samples, significantly improving accuracy of statistical sampling

List of Independent ML Projects:

1. Predicting median household income of counties
    a. Downloaded and cleaned data from Census Bureau Data to analyze and predict county level economic data
    b. Conducted extensive data exploration with different economic factors, examining and constructing regression between various economic indicators
    c. Developed linear regression models to predict median household incomes, informing future stakeholders about counties with potential growth indicators
2. Predicting Stock Prices with RRN
    a. Implemented YFinance API to download and locally store stock data to implement time series analysis on stock data
    b. Implemented linear regression, XGBoost, and RNN using Tensorflow API to create predictive models of stock prices
    c. Improved predictions from previous models using LSTM models
3. Fake News Classifier Using NLP
    a. Extracted and cleaned online news articles from multiple online sources, using BeautifulSoup API, combining multiple available datasets to increase size of samples
    b. Achieved classification accuracy ~92% (recall and F1 scores are similar) using dense neural and LSTM networks
    c. Increased accuracy score of classifier to ~95% by implementing XGBoost algorithm
4. Predicting the Shapes of Galaxies Using Computer Vision
    a. Developed independent Decision Trees, RandomForest, and neural networks using Tensorflow API to classify the

morphologies of galaxies from datasets with ~95%

b. Implemented image detections techniques using SciKit-Image and OpenCV to explore and analyze features that enhance discriminatory features for classification improving prediction accuracy by ~5%, showing drive to investigate feature optimization

5. Implemented UI Friendly Python Interface to create and display complex neural networks
6. Resource Allocator For Group
    a. Given a test simulation and expected scaling relation, predicts wallclock time and resource consumption, allowing for quick adjustments to computational resource allocation

Education:

Ph.D in Physics, University of California, Riverside — Expected Graduation: December 2024

Specializations: Astrophysics, Cosmology, Galaxy Formation and Evolution, Big Data

B.S. Astrophysics, Rutgers University — Graduation Date: May 2018

Minors: Computer Science and Mathematics

Additional Certificates:

- Science 2 Policy Certificate for Technical Writing and Science Communication
  University of California, Riverside
- Unsupervised Learning, Recommenders, Reinforcement Learning
  DeepLearning.AI and Stanford University