

Prabhath Vinay Vipparthi

973-418-9427 | vipparthi.prabhathvinay23@gmail.com | Harrison, NJ | [LinkedIn](#) | [GitHub](#) | [Portfolio](#)

SUMMARY

MS Data Science graduate from NJIT (GPA 3.7, May 2026) with production experience building NLP classification systems, machine learning pipelines, and large-scale data workflows. Built an explainable NLP classification system (FastAPI, spaCy) with full decision traceability, governance logging, and a 350+ test validation suite; benchmarked 6 ML models with SHAP explainability on a 20,000-record credit dataset; and engineered an LLM instruction-tuning pipeline (StarCoder2, vLLM) over 30,000+ source files on 5,791 quality-validated instruction-tuning pairs. Skilled in end-to-end ML development from feature engineering and model selection through production deployment and explainability reporting. work authorization: F-1 OPT, STEM-eligible.

EDUCATION

MS Data Science (Computational Track) — NJIT, Ying Wu College of Computing

May 2026 | Newark, NJ | GPA: 3.7

B. Tech Electrical & Electronics Engineering — Lendi Institute of Engineering & Technology

April 2023 | Andhra Pradesh, India

EXPERIENCE

Graduate Data Science Capstone — NJIT – Learning & Development Initiative (LDI)

Jan 2026 – May 2026 | Newark, NJ

- Engineered a production AI-assisted digital badge classification system (FastAPI, React/Vite, spaCy, SQLite) that automates NJIT's institutional taxonomy classification across 3 dimensions (category, type, cognitive level) achieving 100% classification accuracy on 20 real-world badge submissions validated against ground-truth labels from 5 institutional units (LDI, OSIL, Makerspace, NCE, OGI).
- Designed a 4-layer NLP extraction pipeline: 130+ lexicon phrase patterns, 44 regex rules, spaCy Bloom's taxonomy verb extraction, and an LLM stub handling OBv3 JSON, guided forms, and free-text inputs with robust signal coverage across all badge description styles.
- Built a deterministic 3-stage rule engine (19 rules: category → type → cognitive level) with full rule traceability, 8-element plain-English decision explanations per output, confidence scoring, immutable governance audit logs, and human-in-the-loop override workflows requiring written justification ensuring every classification is explainable and auditable.
- Delivered a 351-test automated validation suite (100% pass rate) across unit, integration, and 7 end-to-end workflow scenarios covering NLP extraction accuracy, full taxonomy classification consistency, and metadata ingestion correctness.

Office Assistant — New Jersey Institute of Technology

Sep 2025 – May 2026 | Newark, NJ

- Developed a web-based study room booking system to automate space reservation workflows for departmental staff and students, replacing manual scheduling processes.

PROJECTS

Heart Failure Prediction — Multi-Model Clinical ML Study | Python · Scikit-learn · TensorFlow/Keras · PyTorch · SHAP · 10-Fold Stratified CV

- Built end-to-end binary classification pipeline on 918 patient records (12 clinical features) using 10-fold stratified CV; benchmarked Random Forest, LSTM, and KNN across 15+ metrics per fold (Accuracy, F1, AUC-ROC, Brier Score, TSS, HSS) RF achieved 86.8% accuracy, F1 0.883, AUC-ROC 0.94; LSTM 84.2%; KNN 71.0% Applied SHAP for instance-level feature attribution, generating global importance rankings and per-prediction explanations surfacing ejection fraction, serum creatinine, and age as dominant clinical predictors for clinician-interpretable model outputs.

Loan Approval Prediction — Explainable Credit Risk Modeling | Python, Scikit-learn, TensorFlow, SMOTE, SHAP, GridSearchCV, Pandas

- Constructed a classification pipeline on 20,000 loan applications (36 features, 76.1% class imbalance); applied SMOTE exclusively on training folds to mitigate data leakage; compared 6 models (LR, DT, RF, SVM, KNN, ANN) with GridSearchCV and 5-fold cross-validated tuning across Accuracy, F1 (macro/weighted), AUC-ROC, and Precision-Recall metric.
- Applied SHAP across all 6 model types for model-agnostic explainability; identified CreditScore, AnnualIncome, and DebtToIncomeRatio as top predictors producing compliance-ready feature attribution outputs structured for regulatory reporting contexts.

StarCoder2 Self-Alignment Pipeline — LLM Instruction-Tuning Dataset | Python, TypeScript, Tree-sitter, vLLM, StarCoder2-3B, Hugging Face, Colab

- Implemented the SelfOSSInstruct methodology from the StarCoder2 technical report to construct a high-quality instruction-tuning dataset for code generation; designed and orchestrated a 3-stage pipeline: (1) seed function extraction and quality filtering, (2) concept and instruction generation via LLM, (3) response generation with model-based quality assessment.
- Processed 30,000+ raw TypeScript files from The Stack v2 (Hugging Face) using Tree-sitter AST parsing for syntactic validation and the TypeScript compiler API for type-checking filtering down to 5,791 quality-validated seed functions (~19% yield), each verified for return type correctness, function completeness, and parsability.
- Orchestrated an S→C→I→R instruction chain (Seed → Concepts → Instructions → Responses) using StarCoder2-3B deployed via vLLM batched inference (batch size 32) on a Google Colab T4 GPU; generated 5,791 concept annotations, 5,791 natural-language instructions, and 448 complete instruction-response pairs with per-output quality scoring for downstream fine-tuning use.

TECHNICAL SKILLS

Programming Languages: Python, SQL, R, TypeScript, Java

ML & Statistical Modeling: Scikit-learn, PyTorch, TensorFlow/Keras, SHAP, SMOTE, GridSearchCV, Cross-Validation

NLP & Text Analytics: spaCy, Hugging Face Transformers, vLLM, Regex Pipelines, Phrase Matching, Bloom's Taxonomy

Data Engineering: Hadoop, MapReduce, HDFS, Pandas, NumPy, ETL Pipelines, Tree-sitter

Visualization & Reporting: Tableau, Power BI, Matplotlib, Seaborn

Cloud, DevOps & Infra: AWS (EC2, S3), Google Cloud, Docker, GitHub Actions, FastAPI, REST APIs

Databases: PostgreSQL, MySQL, SQLite

Tools & Environments: Git, Linux, Jupyter, VS Code, React, Colab