Project Title: Problem Statement 15

"Supervised Learning for City-Level IP Geolocation"

Team Name: GEOSTHIRA

Team Members: Kavyashree K

Margaret Sheela C

Prof. Sneha Zolgikar (Internal Mentor)

College: Vemana Institute of Technology, Bengaluru

Table of Contents

Problem Description

- Current IP geolocation methods are mostly database-driven and often become outdated quickly, leading to poor city-level accuracy.
- IP addresses frequently shift across regions due to ISP reassignments, mobile networks, or routing changes, which causes incorrect mapping.
- Existing solutions generally do not provide confidence levels or error bounds, making it hard to assess how reliable a prediction is.
- Special cases such as VPNs, Carrier-Grade NAT (CGNAT), and Anycast introduce additional challenges, as the same IP can appear in multiple locations at the same time.
- Rule-based and static approaches fail to handle rare cities, class imbalance, and dynamic network conditions, limiting their generalization.
- There is a need for a supervised machine learning model that not only predicts the city-level location of an IP address but also outputs confidence scores and an estimated error radius (in kilometers).
- The proposed solution aims to combine IP→city datasets with auxiliary features (rDNS, RTTs, traceroute hints, time zone patterns, etc.) to improve accuracy and robustness.
- Such a system will be more adaptive, transparent, and reliable compared to traditional geolocation databases, making it suitable for real-world applications.

Key Issues

- Database-driven geolocation is outdated and often inaccurate at the city level.
- IP addresses can shift across regions due to ISP changes and mobility.
- No confidence score or error bound in existing solutions.
- VPNs, Anycast, and Carrier-Grade NAT make location prediction unreliable.
- Rule-based methods fail to generalize across the derived features

Solution Proposed

The following points outline the structured approach used in the **GEOSTHIRA IP Geolocation System**, which combines AIORI traceroute data, advanced feature extraction, and supervised learning to provide **city-level IP predictions** with an interactive confidence-based dashboard.

- **DataCollection**: The dataset was sourced from AIORI traceroute measurements across multiple domains to capture diverse, real-time network behavior. To enhance geographic representation, synthetic zones such as *North, North-East, South, and West* were created. The initial raw dataset (~700 samples) contained spikes, missing values, and irregularities, which were preprocessed for reliability.

- **Feature Engineering:** To improve data coverage, synthetic bootstrapping was applied to balance underrepresented regions. Outliers and anomalies were removed using Interquartile Range (IQR) filtering, refining the dataset from ~50,000 raw points to a clean and structured set of 36,000 training samples. Extracted features include RTT statistics, IP class, cluster density, and reverse DNS attributes, representing the unique network signature of each IP.

- **Model Training and Evaluation:** Two feature sets were prepared per IP: one with core statistical features and another enhanced with reverse DNS tokens. The Random Forest classifier was selected after comparative testing with Naive Bayes for its superior accuracy and stable confidence estimations. The trained model was serialized using *joblib* for deployment in both Google Colab and the integrated Gradio environment.

- **Prediction Modes and User Interaction:** The deployed system supports three modes of input — *Single IP*, *Multiple IPs*, and *CSV Upload*. Users can enter IPs manually or upload datasets for batch processing. Each prediction returns the predicted city, confidence score, and error bound estimate, along with a bar chart visualization for multiple IPs. The system automatically detects and flags private IPs and global public DNS IPs (e.g., *8.8.8.8*, *1.1.1.1*), ensuring no invalid or global service IPs are processed for location prediction.

- **Confidence Scoring and Visualization:** For every prediction, the system displays probability-based confidence scores indicating the reliability of the result. A dynamic confidence graph is generated for multi-IP predictions, providing visual insight into the prediction distribution. Although explicit error radius mapping was not implemented, these probabilistic measures ensure transparency in prediction certainty.

- **Web Dashboard and User Authentication:** The complete model is integrated into an interactive Gradio web dashboard that includes *login and signup authentication*, *About section*, and *prediction interface*. The dashboard provides a user-friendly platform for real-time testing and visualization without the need for coding. This front-end serves as an accessible demonstration of AI-driven IP intelligence.

- **Tools & Environment:** The core development and model training were executed in Google Colab, with supplementary analysis in MATLAB for feature validation and comparative experiments. The deployment utilized Python libraries including *pandas, numpy, scikit-learn, matplotlib, joblib,* and *gradio*. The graphical outputs such as confidence bar charts and data tables help users interpret prediction outcomes effectively.

Optimization Proposed by the Team

Our team identified limitations in baseline IP geolocation approaches and proposed optimizations to improve **accuracy, confidence, and robustness** of city-level predictions. The goal was to make the system practical for real-world use while remaining aligned with the features and constraints of our collected dataset.

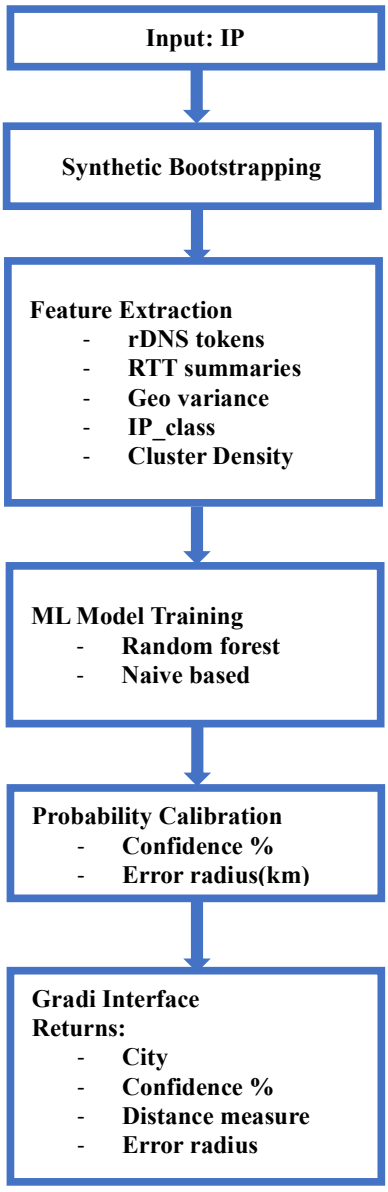
Key Optimizations and Enhancements

- **Enhanced Feature Set:** Extracted key network features such as **avg/min/max RTT, RTT range, RTT ratio, cluster density, geo-variance, temporal activity, and reverse DNS tokens**. These features help the model capture IP differences across zones, improving city-level predictions.
- **Data Balancing & Probability Calibration:** Handled dataset imbalance across zones using **synthetic bootstrapping** and **IQR filtering**, reducing noise and producing a clean, balanced dataset of **36,000 samples** suitable for training.
- **Model Seclection and Evaluation:** Trained both **Naive Bayes** and **Random Forest** models using the cleaned dataset. **Random Forest** provided higher accuracy and confidence, so it was chosen as the final model. Evaluation was done via **train-test split (80-20)**, with accuracy and confusion matrix as primary metrics.

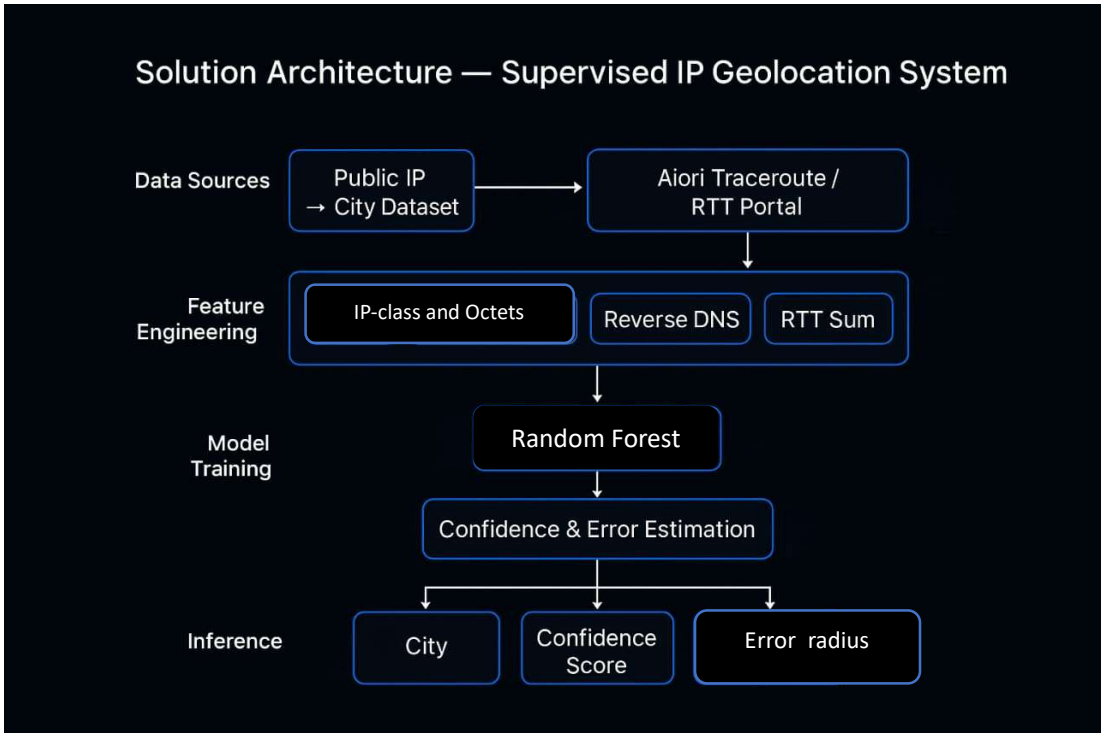
Comparison: Before vs. After Optimizations

Aspect	Before (Baseline Methods)	After(Proposed Optimization)
Accuracy (City-level)	Often outdated, low precision	Improved using Random Forest trained on richer, feature-enhanced dataset
Data Handling	Imbalanced (big zones dominate)	Balanced using synthetic bootstrapping and IQR filtering
Confidence Output	Not available	Probability-based confidence scores
Error Bound	Absent	Not implemented (future scope)
Special Cases	Misleading for unknown IPs	Predictions can still be made for new IPs using simulated features
Output	Single city guess	Prediction for single or multiple IPs with confidence scores

Flow Chart of Optimization Process:

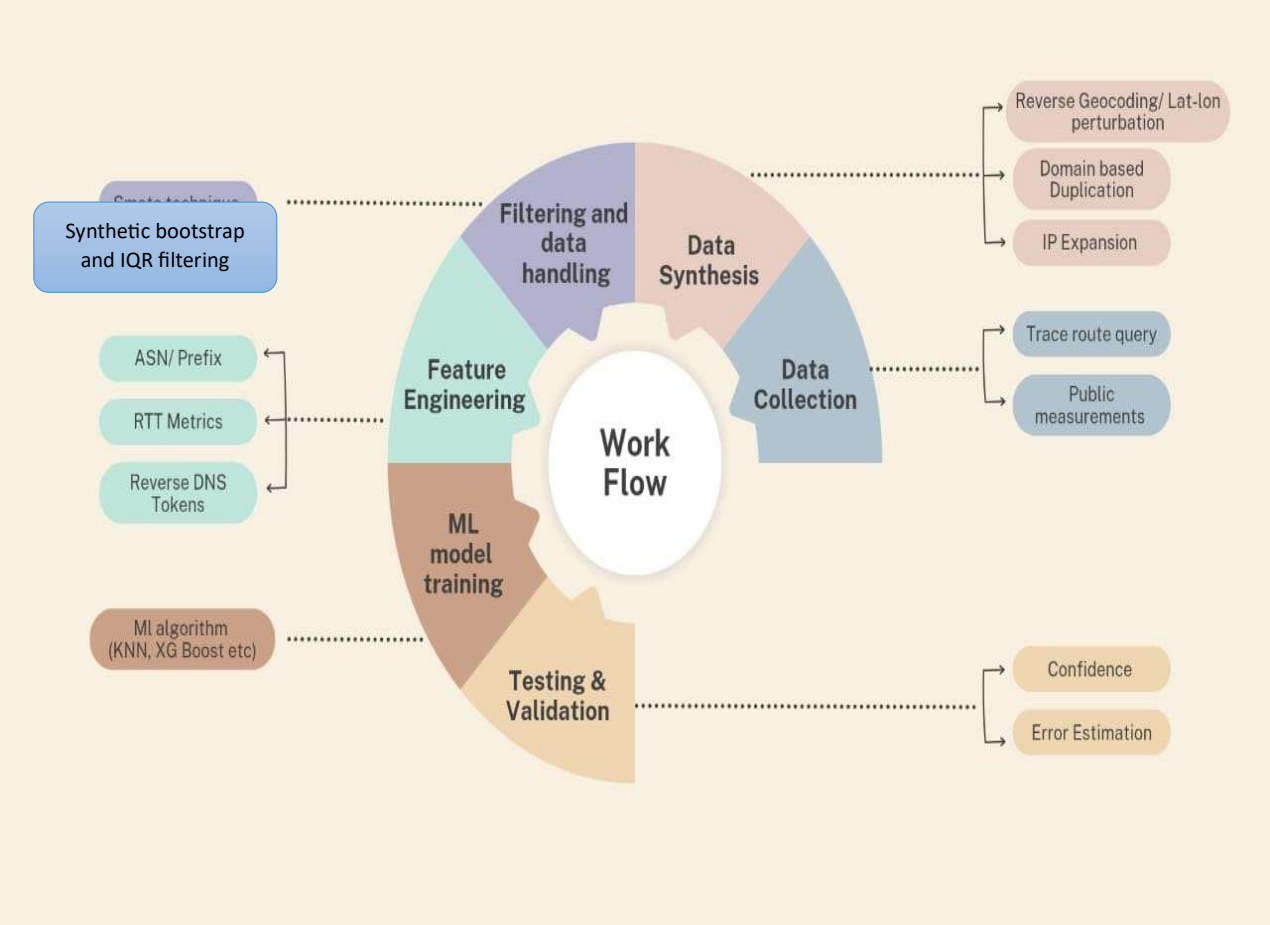


Solution Architecture and Design



1. The system uses live traceroute measurements from AIORI across multiple domains and synthetic zones (North, North-East, South, West) instead of static IP databases. This ensures the model learns from real network behavior.
2. Each IP is represented as a feature profile, including RTT statistics, reverse DNS patterns, IP distance, geo-variance, cluster density, and other derived features. These features allow the model to map IPs to cities effectively.
3. The Random Forest model is trained to predict the most likely city and provide a probability-based confidence score for each prediction, helping differentiate well-behaved IPs from uncertain cases.
4. The system supports single IP and multiple IP (CSV) predictions, generating confidence scores for each input to guide decision-making.
5. The architecture is modular and scalable, allowing future improvements in feature extraction, model training, or inclusion of additional network measurements without major redesign.

Workflow Diagram



1. Started with raw traceroute/IP data instead of static databases to ensure the system reflects real Internet behavior rather than outdated registries.

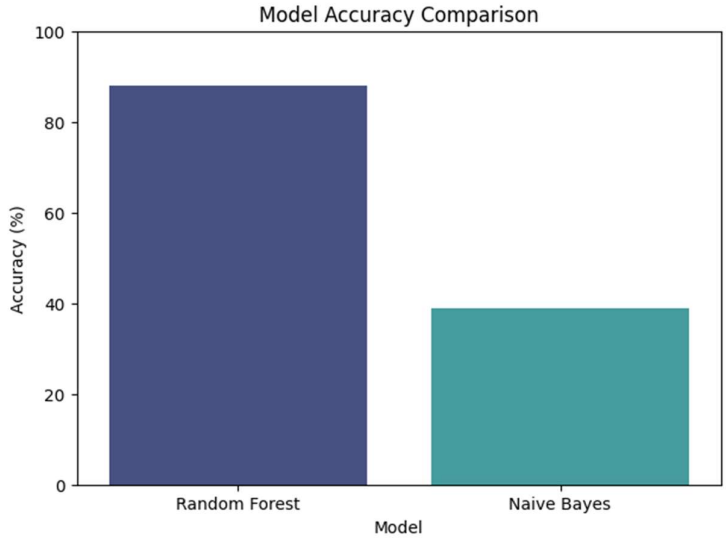
2. Added filtering and synthetic expansion early so that bad measurements don't mislead the model and underrepresented regions still get learned.

3. Moved to feature engineering only after stability, because extracting patterns from noisy RTTs or missing ASNs would distort learning.

4. Chose lightweight ML models first (KNN/XGBoost) so we could validate feasibility quickly before overcomplicating the stack.

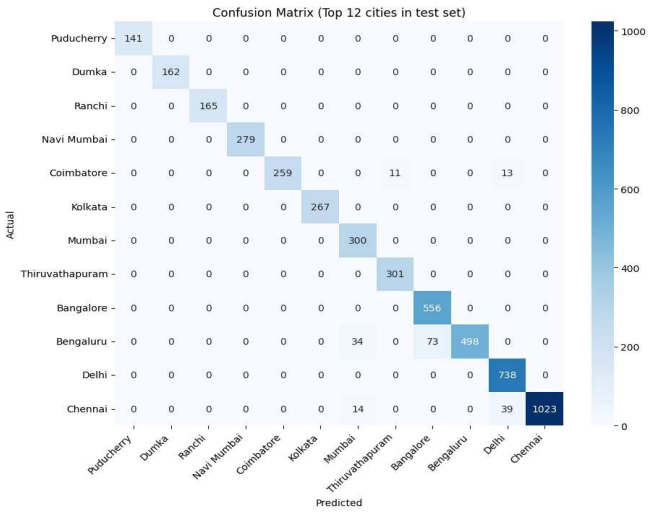
5. Ended with confidence/error estimation instead of only accuracy, since geolocation is probabilistic by nature and practical use cases demand reliability, not just predictions.

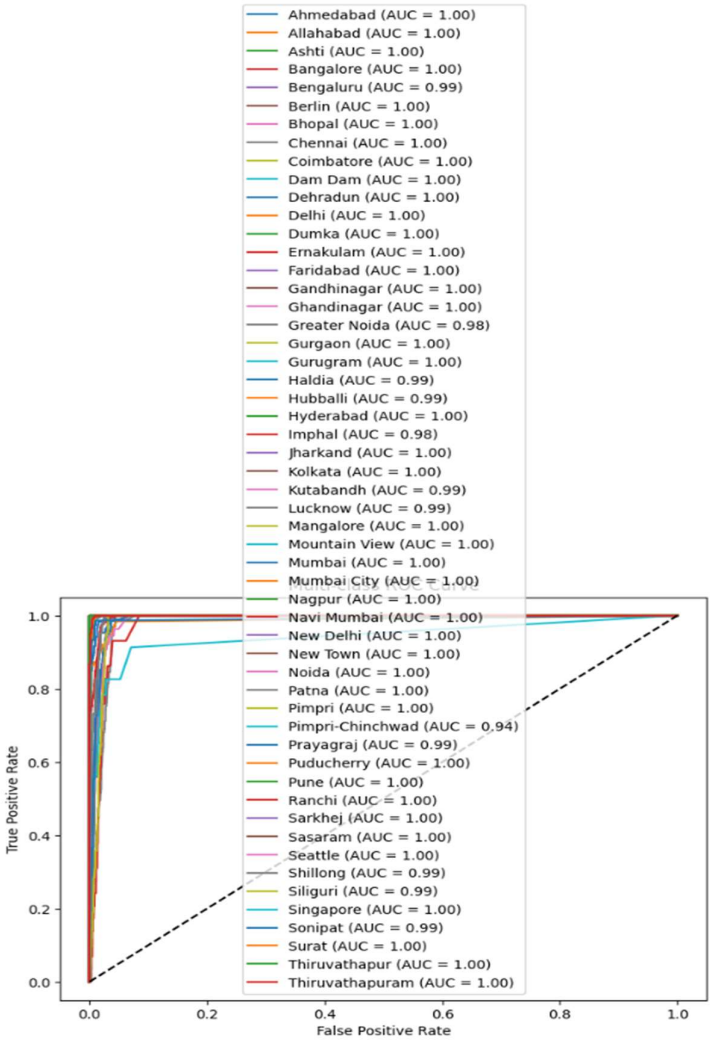
OUTPUTS OBTAINED:



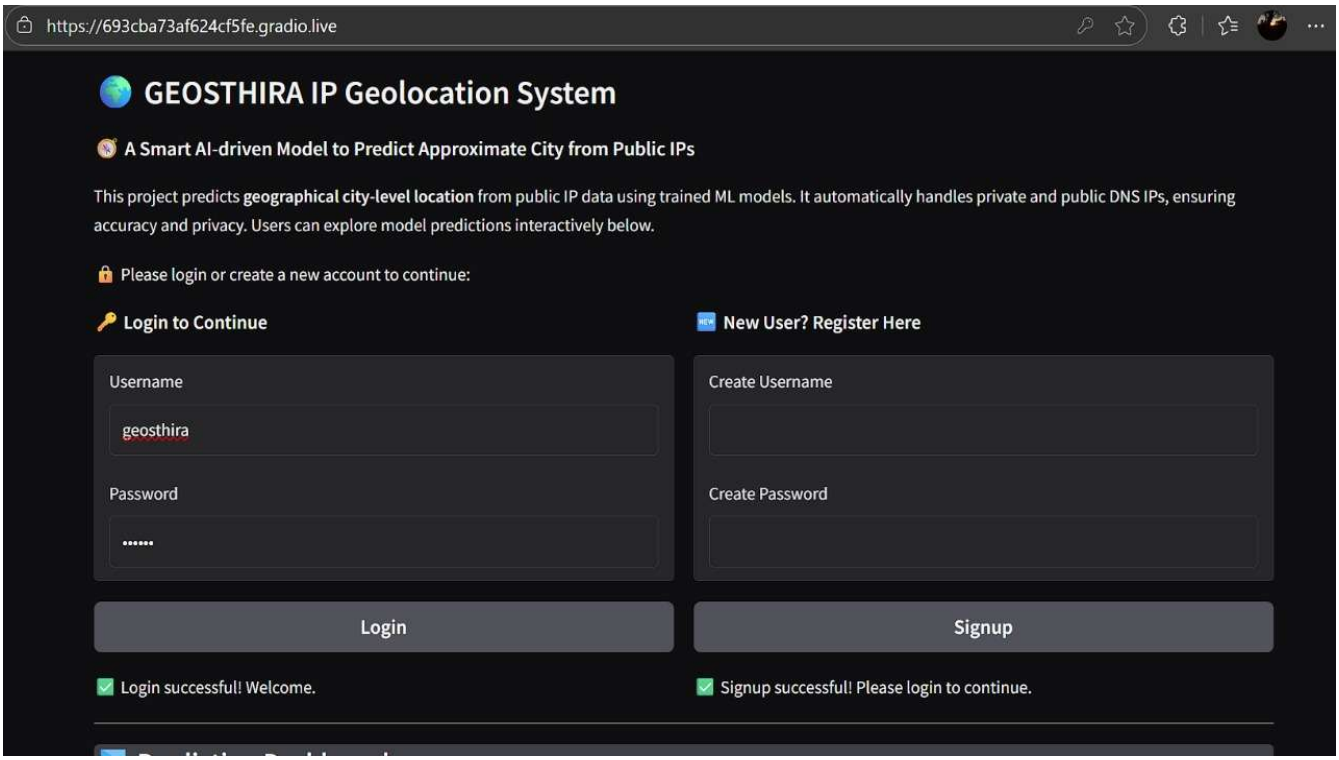
📈 MODEL COMPARISON SUMMARY:

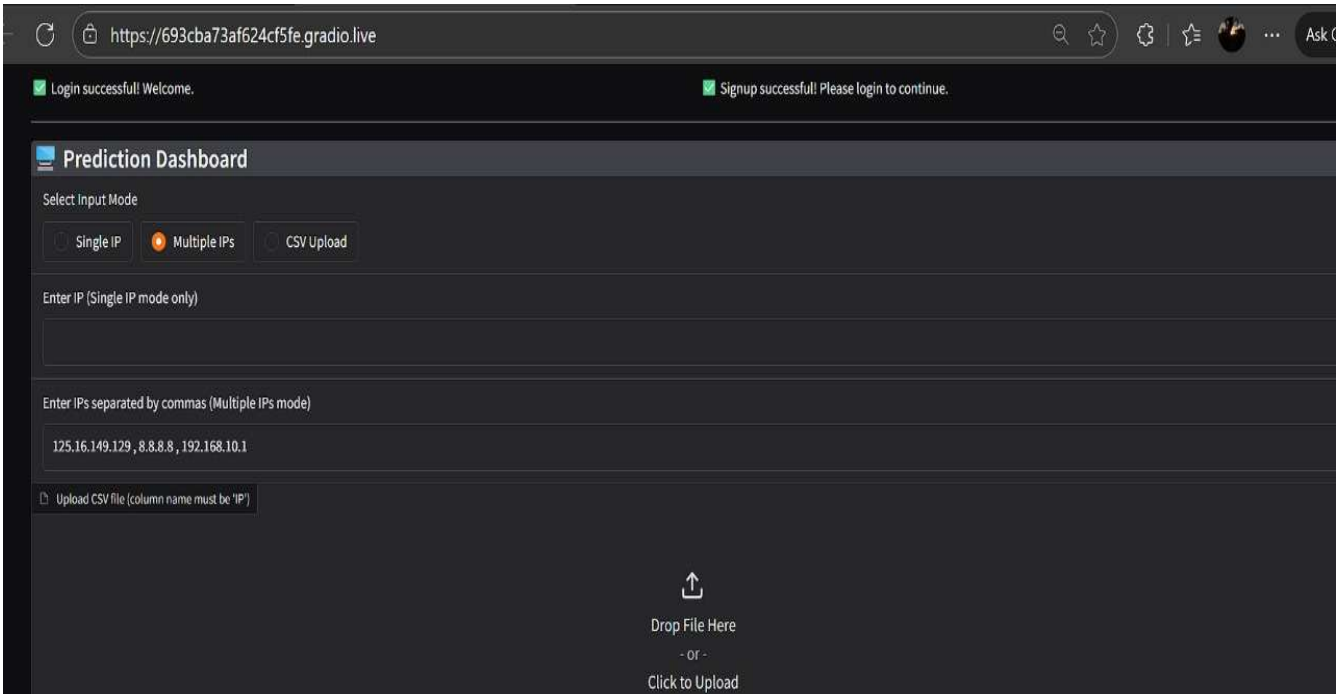
	Model	Accuracy (%)
0	Random Forest	87.864802
1	Naive Bayes	38.767477

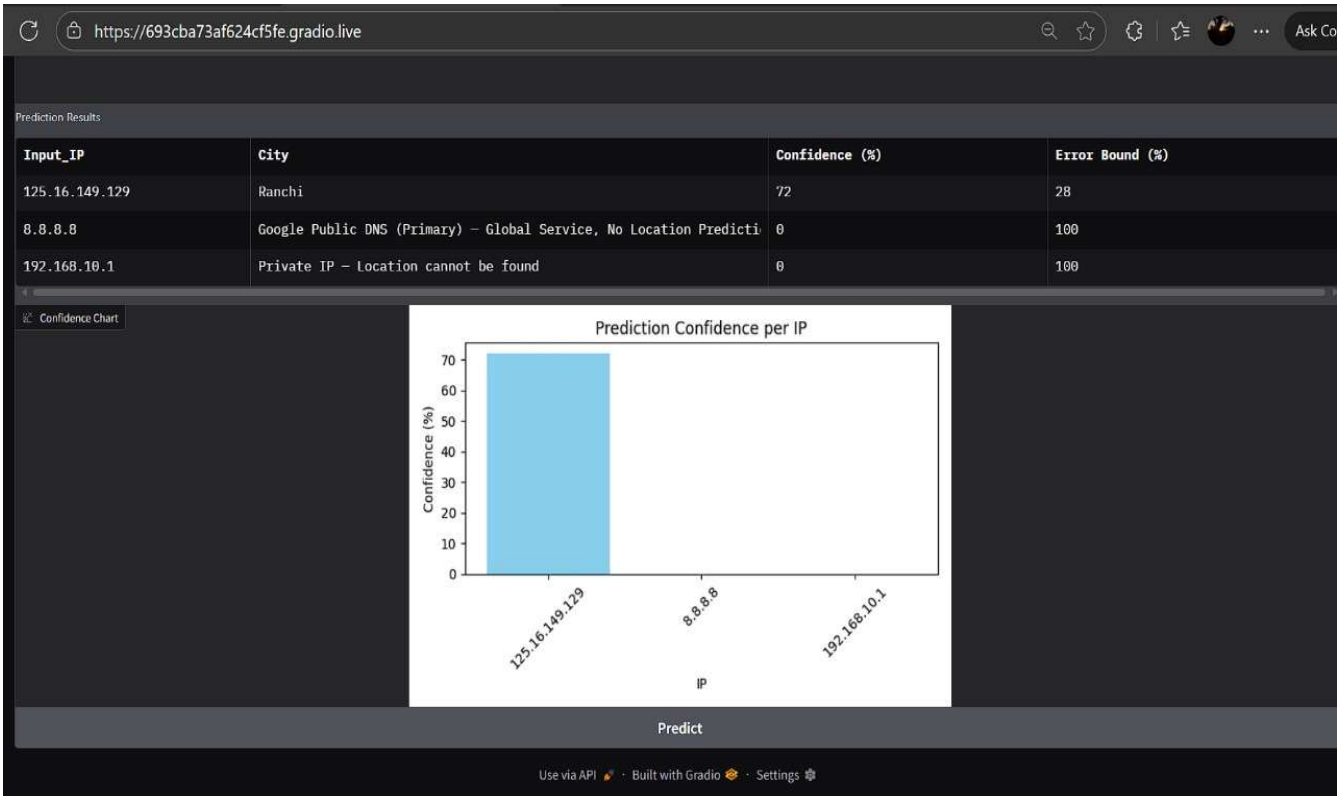




GeoSthira IP Geolocation system Dashboard:



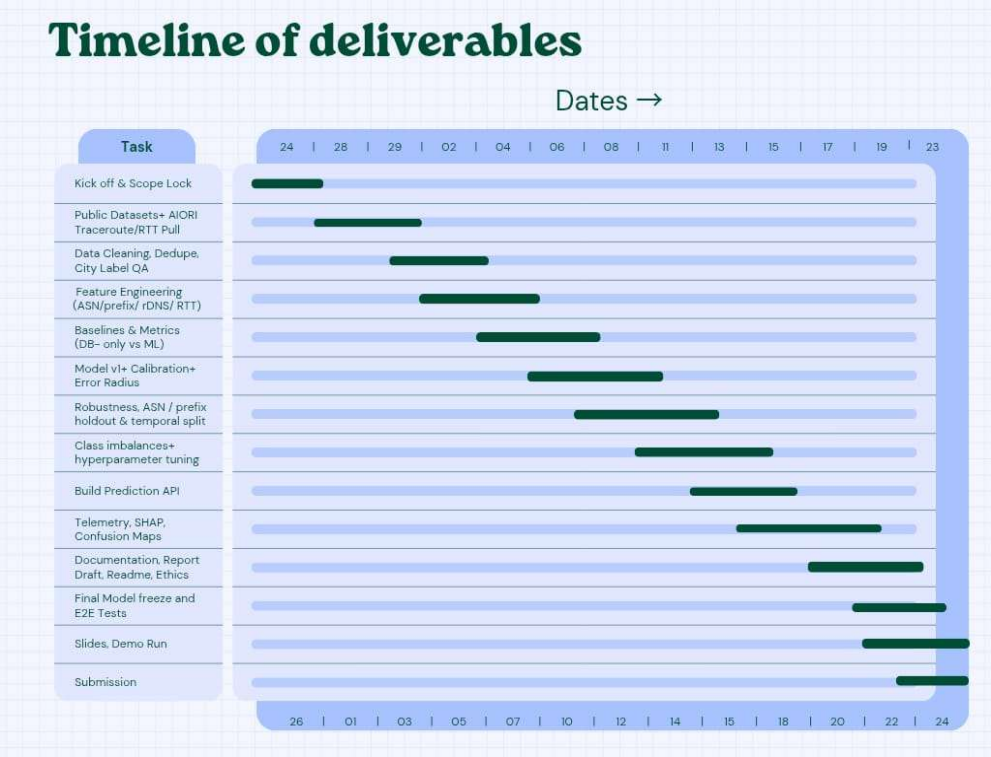




Result Analysis

1. The trained Random Forest model predicts the approximate city for a given public IP address using extracted numerical and network-based features.
2. The model automatically filters private IPs and public DNS addresses like Google or Cloudflare to avoid false predictions.
3. The output table displays predicted city, confidence percentage, and error bound for each input IP, with a visual bar chart for confidence comparison.
4. The system provides an interactive Gradio dashboard with login/signup authentication where users can test single, multiple, or CSV-based IP predictions easily.

Timeline of Delivery:



References

1. IEEE Journals and Papers on IP Geolocation Techniques
   - Provides foundational knowledge and methods for accurate IP geolocation.
2. ACM SIGCOMM Publications on Internet Mapping, DNS, and Traceroute
3. AIORI Portal – Network and IP Data Access
   - Primary data source for IP addresses, zones, and network measurements in our project.
4. Google – IP Geolocation, DNS, and Networking Tools
   - Useful for verifying geolocation results and learning practical implementation methods.
5. Python Documentation – Data Analysis, Machine Learning, and Feature Engineering
6. MATLAB Documentation – Signal Processing, Machine Learning, and AI Toolboxes
   - Used for experimenting with algorithms, model training, and visualization of results.
7. Jupyter Notebook – Interactive Computing and Experimentation Environment
   - Provides an environment to organize code, run experiments, and document findings.

Conclusion

The “Supervised Learning for City-Level IP Geolocation” project, titled GEOSTHIRA IP Geolocation System, provides an intelligent and interactive approach for predicting city-level locations of public IPs. Unlike static database-based lookup methods, this model uses machine learning with real-time traceroute features to achieve higher adaptability and accuracy. The dashboard allows users to test single or multiple IPs or upload datasets, while automatically filtering private and global DNS IPs to ensure valid and ethical predictions.

The integration of a Random Forest model enables stable and explainable predictions with confidence and error-bound metrics. The Gradio web dashboard enhances usability by combining model prediction, authentication (login/signup), and visual output in a single platform. This makes the system suitable for network research, educational demonstrations, and cybersecurity analysis.

The project successfully demonstrates how AI-driven geolocation can provide meaningful insights about public IP distributions. With future improvements — such as live data integration, expanded city coverage, and real-time visualization dashboards — this model can evolve into a fully deployable real-world IP intelligence service for smart network applications.

Call to Action

We encourage continued enhancement of this system by integrating real-time IP feeds, refining feature engineering with dynamic latency and traceroute metrics, and deploying API-based services for broader accessibility. The GEOSTHIRA model proves that supervised learning can make IP geolocation more reliable, transparent, and scalable for both academic research and industry use.