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- Current IP geolocation methods are mostly database-driven and often become outdated quickly, leading to poor city-level accuracy.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- Database-driven geolocation is outdated and often inaccurate at the city level.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 aims to predict the Indian city of an IP address using machine learning, without depending on any external geolocation databases. The improved system integrates FastAPI deployment, WHOIS-based enrichment, IPAPI verification, and feature-distance similarity analysis to enhance prediction accuracy and reliability.

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 — one with statistical features and another enhanced with reverse DNS tokens. The **Random Forest classifier** was chosen over Naive Bayes for higher accuracy and stable confidence estimation. The trained model was serialized using **joblib** for deployment.

- **Prediction Modes and User Interaction:** The system supports **Single IP**, **Multiple IPs**, and **CSV Upload** modes. Each prediction returns the city, confidence score, and error bound, along with a visualization for multiple IPs. Private and global DNS IPs (e.g., *8.8.8.8, 1.1.1.1*) are automatically filtered.

- **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 FastAPI-based web dashboard that includes login and signup authentication, an About section, and a prediction interface. The dashboard provides a user-friendly platform for real-time testing and result visualization. This front end serves as an accessible demonstration of AI-driven IP intelligence using FastAPI integration.

- **Tools & Environment:** The core development and model training were executed in Google Colab, and the deployment utilized Python libraries such as *pandas*, *numpy*, *scikit-learn*, *matplotlib*, and *joblib*, integrated with FastAPI for real-time interaction. The graphical outputs, including confidence bar charts and data tables, help users interpret prediction outcomes effectively.

Optimization Proposed by the Team

Our team identified key limitations in baseline IP geolocation methods and introduced targeted optimizations to enhance the **accuracy, confidence, and robustness** of city-level predictions. The objective was to make the system more practical for real-world use while staying consistent with the available dataset and its constraints.

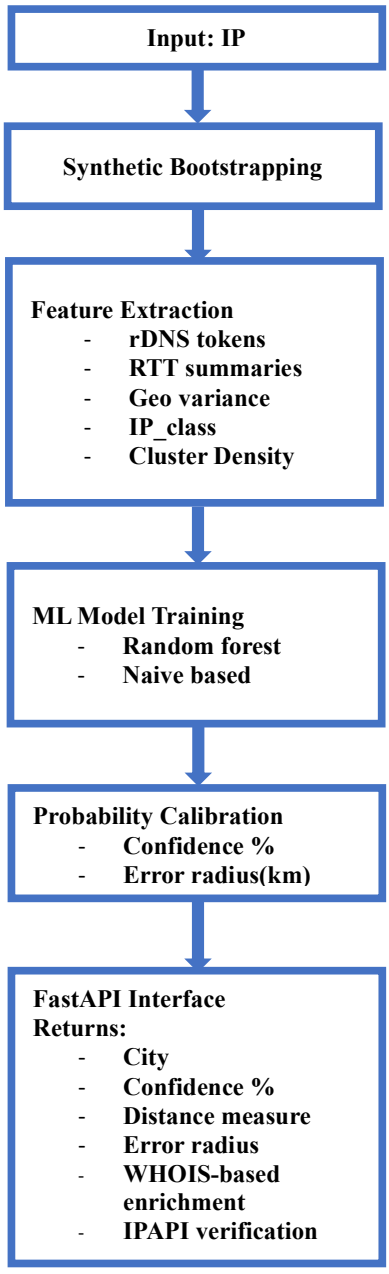
Key Optimizations and Enhancements

- **Enhanced Feature Set**: Extracted critical network features such as average/min/max RTT, RTT range, RTT ratio, cluster density, geo-variance, temporal activity, and reverse DNS tokens. These collectively help the model differentiate IPs across regional zones, improving prediction precision.
- **Data Balancing & Probability Calibration:** Addressed dataset imbalance using synthetic bootstrapping and IQR-based filtering, which reduced noise and produced a balanced dataset of 36,000 clean samples ready for model training.
- **Model Seclection and Evaluation:** Trained both Naive Bayes and Random Forest models on the refined dataset. Random Forest achieved higher accuracy and confidence stability, thus selected as the final model. Evaluation was conducted using an 80–20 train-test split, with accuracy and confusion matrix as key performance metrics.

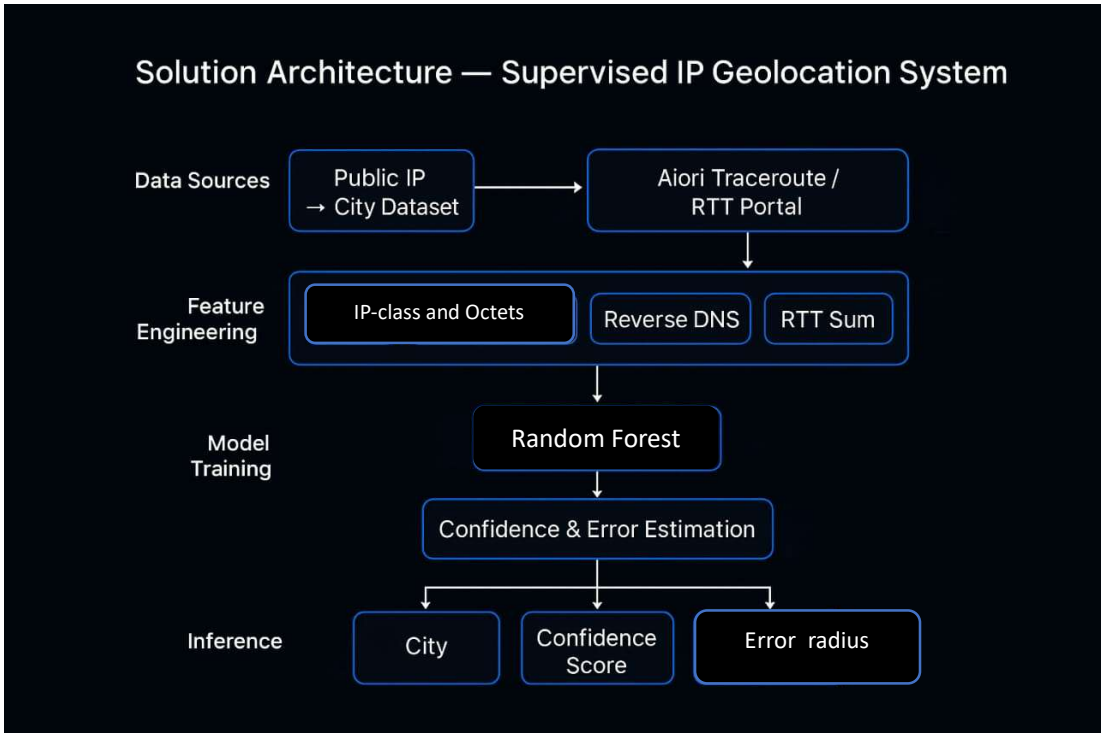
Comparison: Before vs. After Optimizations

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:

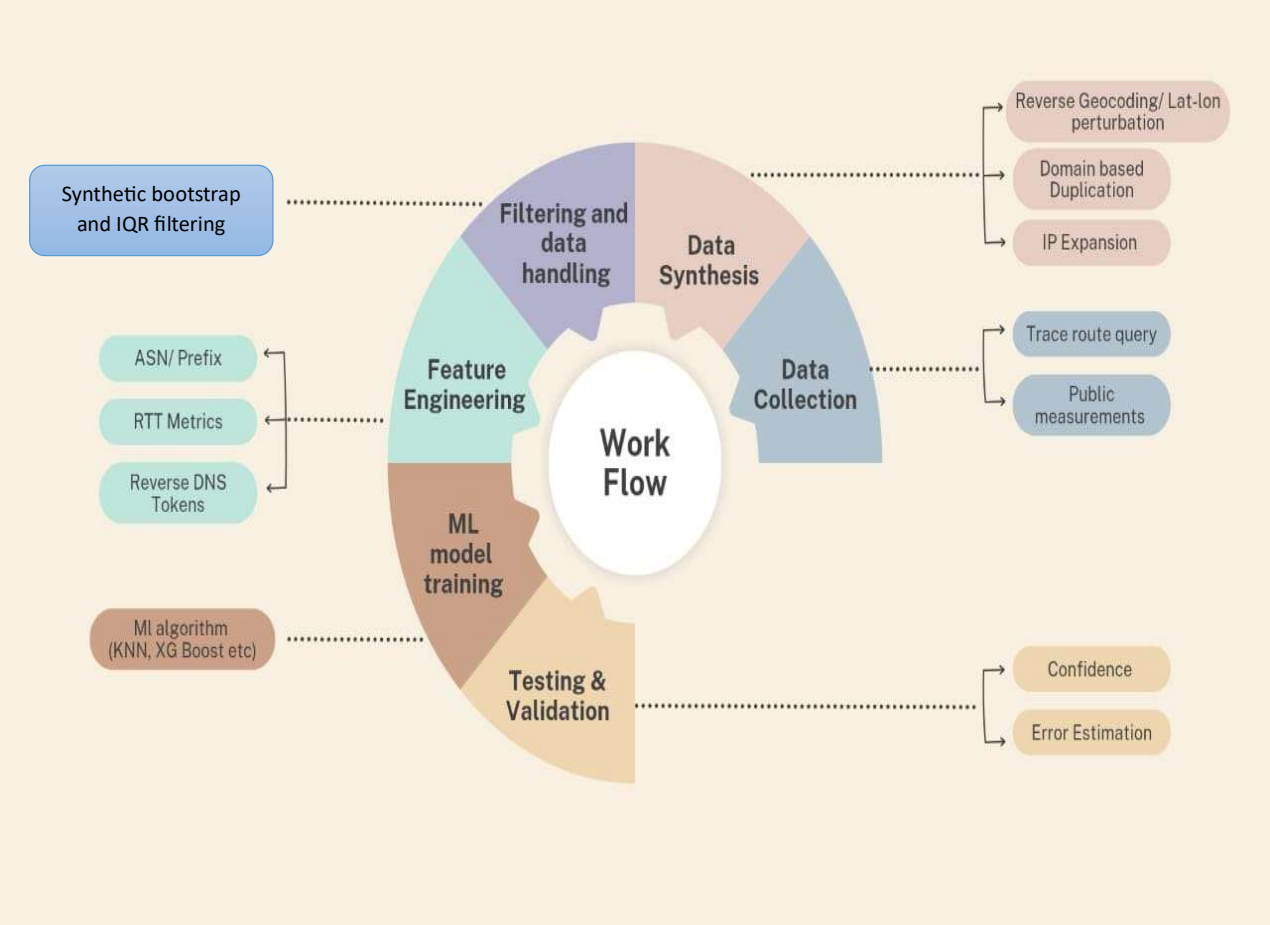


Solution Architecture and Design



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.

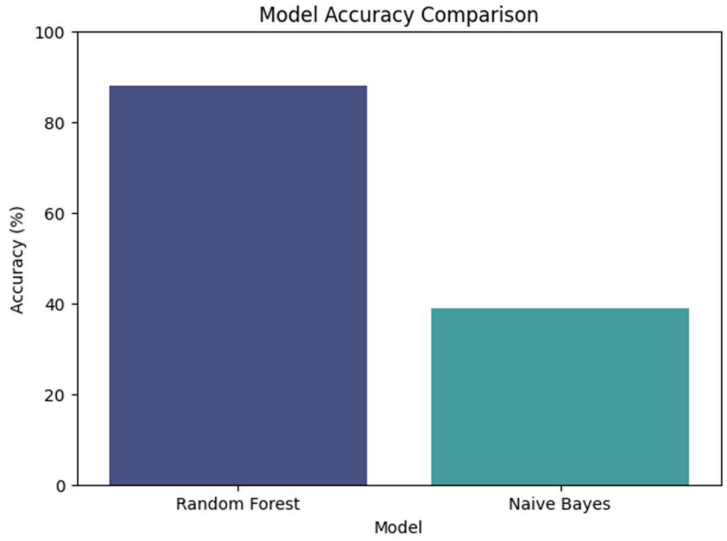
2. Added filtering and synthetic expansion early so that bad measurements don't mislead the model and underrepresented regions still get learned.

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.

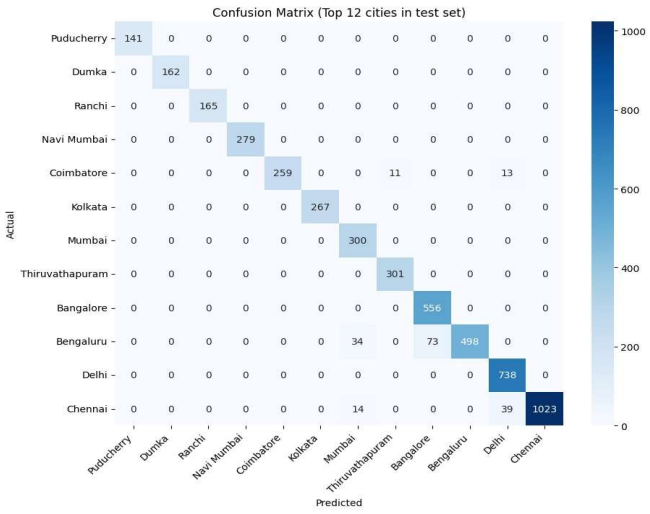
5. Ended with confidence/error estimation instead of only accuracy, since geolocation is probabilistic by nature and practical use cases demand reliability, not just predictions.

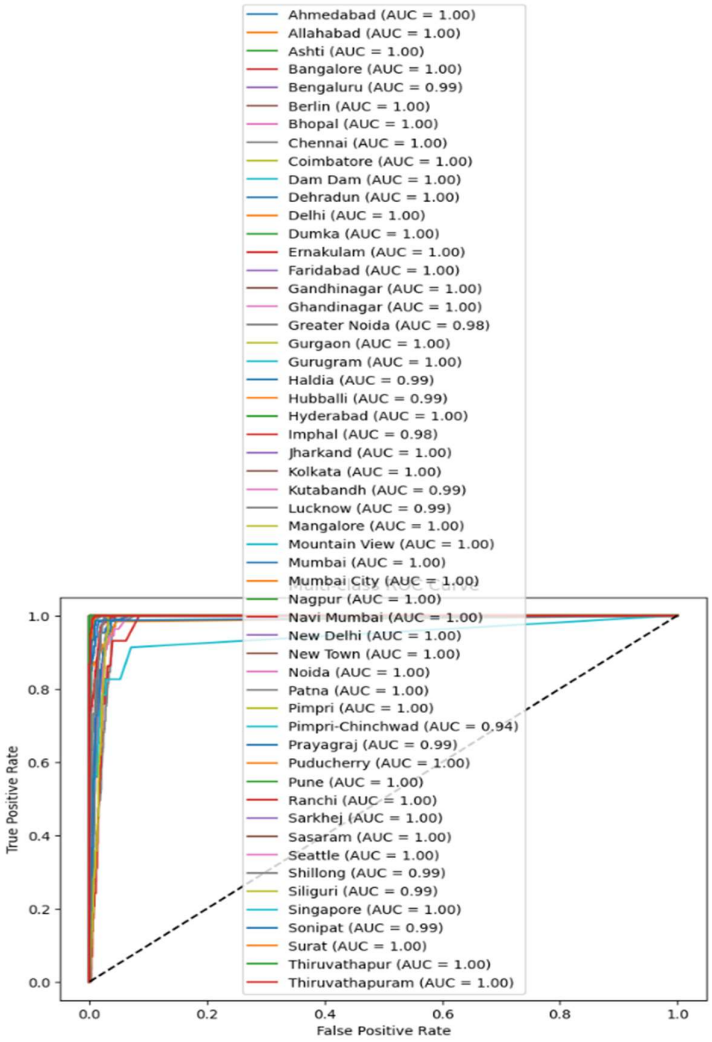
OUTPUTS OBTAINED:



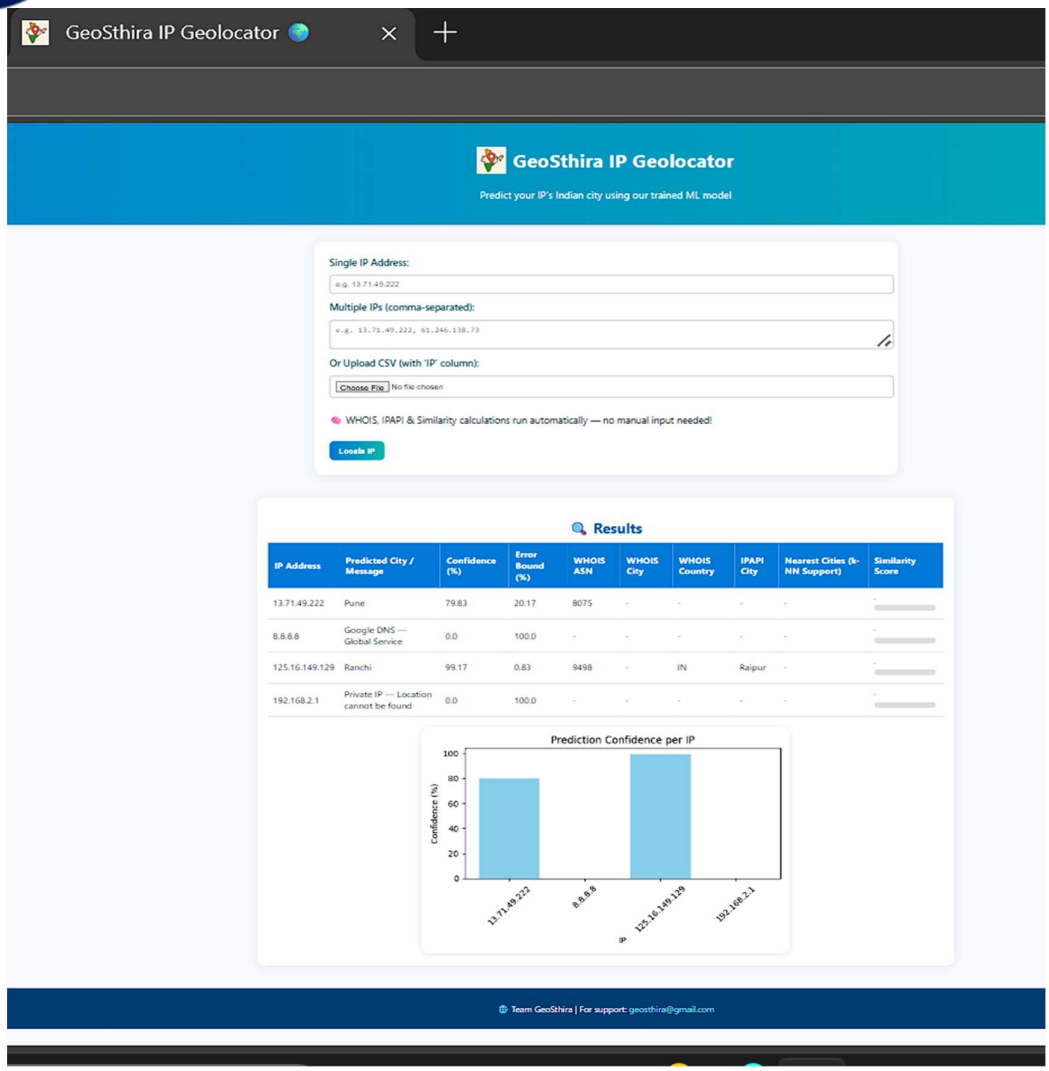
📈 MODEL COMPARISON SUMMARY:

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





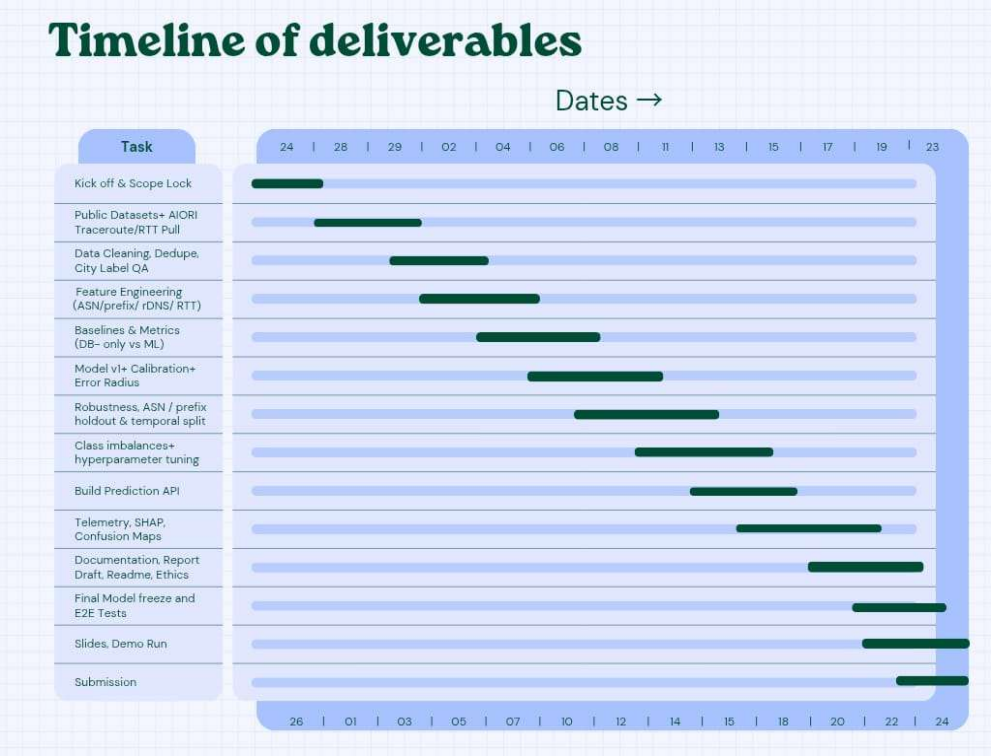
GeoSthira IP Geolocation system – User Interface:



Result Analysis

1. The trained Random Forest model predicts the most probable Indian city for a given public IP address based on extracted numerical and network-level features such as RTT statistics, IP class, and reverse DNS tokens.
2. The system automatically detects and filters private IPs and global public DNS addresses (e.g., Google 8.8.8.8, Cloudflare 1.1.1.1) to prevent false or irrelevant predictions.
3. The output table displays the predicted city, confidence percentage, distance similarity measure, and estimated error bound for each input IP, along with a confidence bar chart for easy visual comparison.
4. The model is deployed via an interactive FastAPI dashboard with authentication support, allowing users to test single, multiple, or CSV-based IP predictions in real time.

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 and Call to Action

The GEOSTHIRA IP Geolocation System demonstrates the potential of supervised machine learning in predicting city-level locations of Indian public IP addresses. Unlike traditional database-based lookup methods, this system uses traceroute latency and network-derived features, allowing it to adapt dynamically and provide accurate, explainable predictions. The model is trained using a Random Forest algorithm, offering stability, interpretability, and confidence-based outputs for every prediction.

Through an integrated FastAPI backend and interactive dashboard, GEOSTHIRA provides users with a seamless interface to test single or multiple IPs, or even upload CSV datasets. The system automatically filters out private IPs and public DNS servers (like Google or Cloudflare) to maintain ethical and valid geolocation results. Each output includes details such as Predicted City, ISP/Organization, WHOIS Data, IP Type, and Model Confidence, ensuring both technical depth and user clarity.

This intelligent model holds strong potential for cybersecurity, network analysis, and academic research, where IP-level location insights can improve transparency, traffic management, and security response strategies.

Looking ahead, GEOSTHIRA can be enhanced by integrating real-time IP data streams, expanding city coverage, and incorporating dynamic traceroute metrics for even higher prediction precision. Future deployment as a REST API service with live visualization dashboards will make it a powerful, scalable tool for real-world IP intelligence applications.

In essence, GEOSTHIRA proves that AI-driven geolocation can make IP intelligence more accurate, transparent, and future-ready, bridging the gap between data science and network engineering.