Seasonal Pollution Episode Prediction

Nishal Save · Shantanu Ghaisas · Sanmati Sawalwade

CHALLENGE

The Problem

North Indian cities face severe seasonal pollution crises every winter, yet warnings often come too late for effective public health response.

🔥

Stubble Burning

Millions of tons of crop residue burned in Punjab & Haryana during Oct-Nov, creating toxic smoke

🌡️

Meteorological Traps

Winter inversions, low wind speeds, and high humidity trap pollutants at ground level

🎇

Festival Emissions

Diwali fireworks add massive particulate matter spikes during already poor air quality

⏱️

Reactive Systems

Current systems react after pollution spikes. Predictive capabilities needed for proactive action

Our Goal: Build a machine learning system that predicts severe pollution episodes before they occur

Data Collection

APPROACH

Dataset Overview

200	19	7	5 yrs
Records	Features	Cities	2019-23

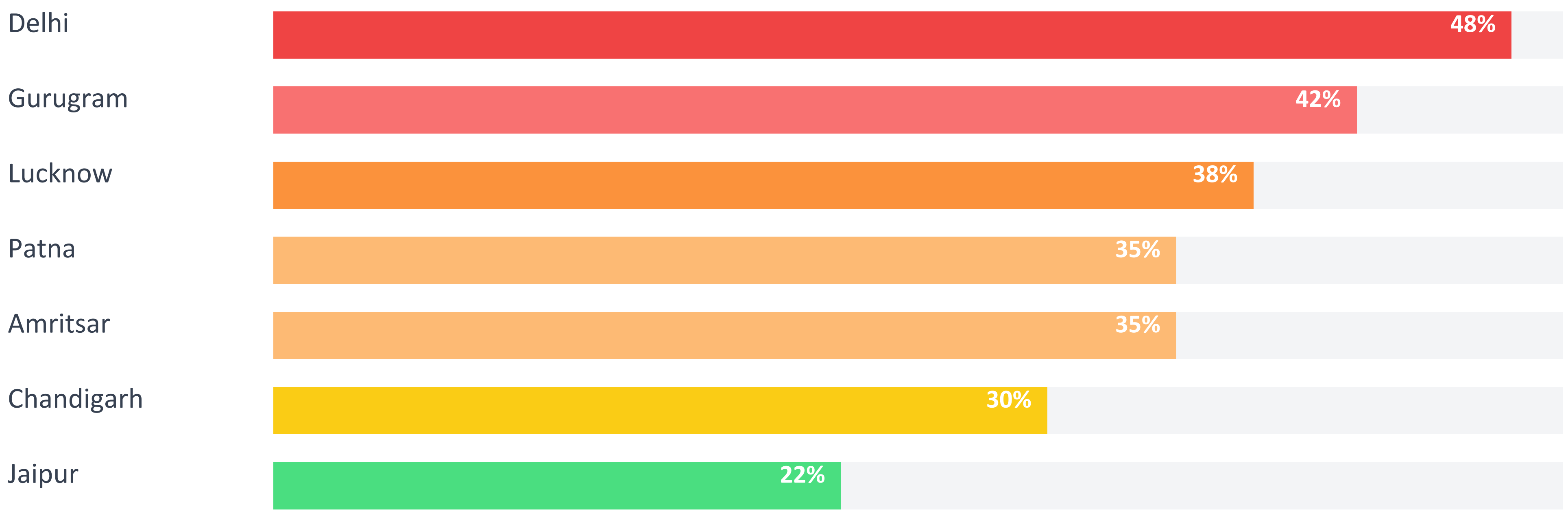
Target Distribution

🔵 Normal: 55.5% 🟠 Severe: 44.5%
✓ Well-balanced, no resampling needed

🎯 5% Random Noise Injection

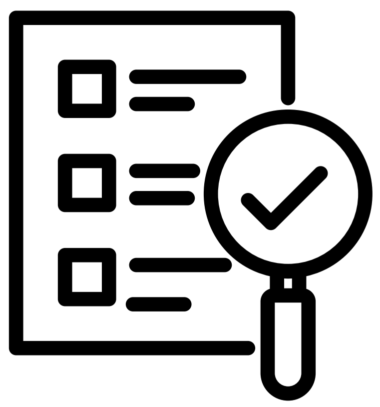
Labels deliberately flipped to simulate real-world uncertainty and prevent overfitting

City-wise Base Severity Probability



Data sources

- NASA FIRMS (satellite fire data)
- OpenWeatherMap (pollution)
- Kaggle CSV file



PIPELINE

Data Generation & Feature Engineering

1. Setup
Libraries
7 cities
Diwali dates

→

2. Generate
Weather
Fire counts
AQI history

→

3. Engineer
5 derived features
Domain knowledge

→

4. Clean
Imputation
Outlier clipping

Raw Features (10)

temperature_c, humidity_pct
wind_speed_kmh, wind_direction_deg
fire_count_punjab, fire_count_haryana
previous_day_aqi, consecutive_poor_days
month, days_from_diwali

→

Engineered Features (5)

weather_dispersion_index
= wind × (1 - humidity/100)

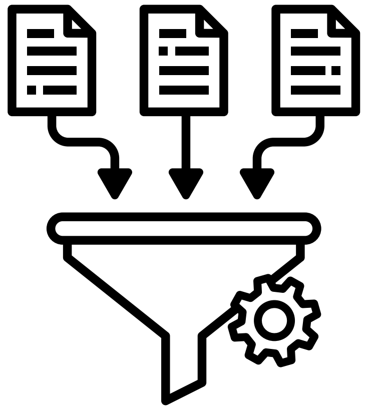
stubble_impact_score
*= (punjab + haryana) × wind_bonus**

pollution_momentum
= AQI × (1 + consecutive×0.06)

diwali_impact_zone
= 1 if within 5 days of Diwali

inversion_risk
= 1 if temp<15°C AND humidity>70% AND wind<10

Severity Prob = city_base + weather(low wind +10%) + fires(>1100: +12%) + AQI(>350: +15%) + diwali(±2 days: +12%) + season(Nov: +6%)



DELIVERABLES

Outputs & Summary

📁 Output Files

pollution_episode_dataset.csv
14 columns (base)

pollution_dataset_with_features.csv
19 columns (+ engineered)

📊 Visualizations

- Target distribution
- Correlation heatmap
- Box plots & scatter plots
- Monthly patterns

✅ Data Quality & Validation

200	0
Total Records	Missing Values

Cleaning: Median imputation, duplicate removal | Outlier Bounds: Temp 0-45°C, Wind 0.5-45 km/h

🎯 Key Design Decisions

✓ Domain-driven severity formula ✓ Overlapping distributions for realistic ML
✓ 5% noise injection to prevent overfitting ✓ Reproducible pipeline (seed = 42)

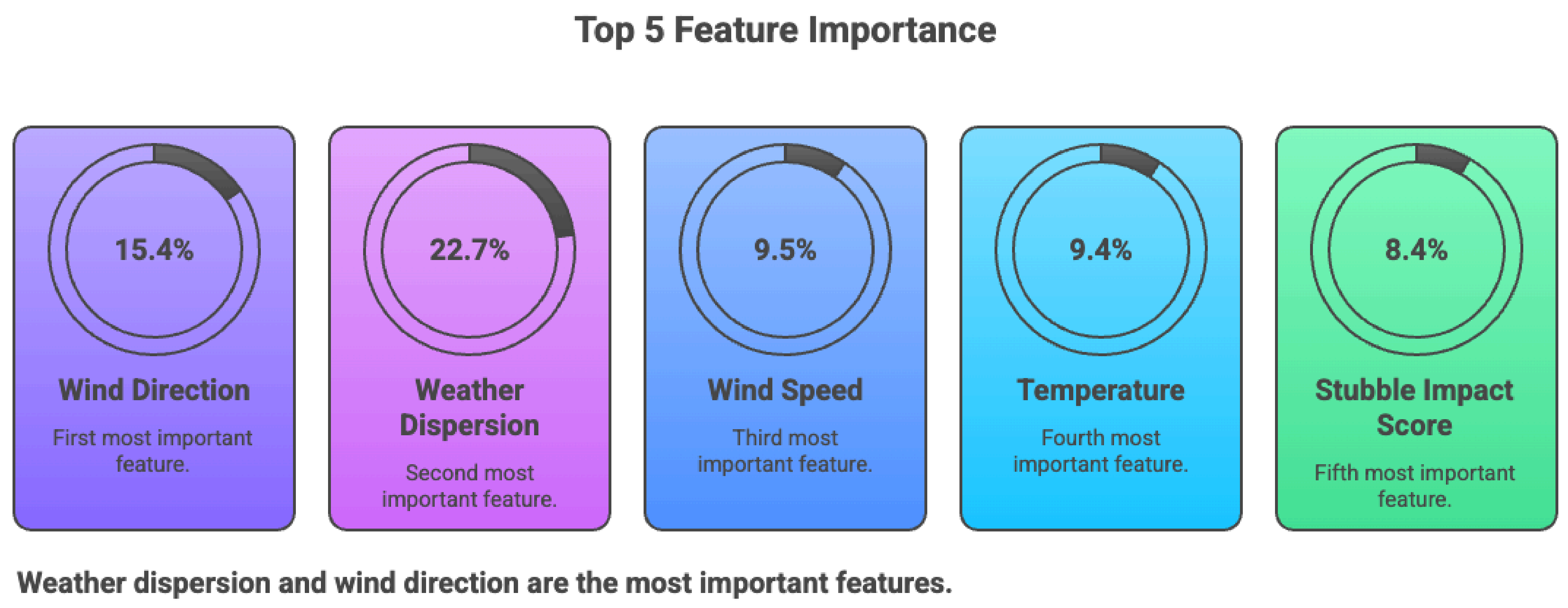
Next Step: Model Training → Train Decision Tree, Random Forest, and XGBoost classifiers on this dataset

Model Training and Evaluation

Phase 1: Feature Analysis and Data Preparation

Train - Test Split [Stratified 80/20]

- Training: 160 samples (71 severe)
- Testing: 40 samples (18 severe)
- Balance: Preserved in both sets



Key Insight 1: Weather conditions dominate predictions (40% combined importance), with wind direction alone accounting for 15.4%—atmospheric dispersion capability matters more than pollution sources.

Key Insight 2: Engineered features validated our domain expertise—pollution_momentum, stubble_impact_score, and weather_dispersion_index all ranked in top 6, proving contextual impact beats raw fire counts.
agraph text

EXECUTION

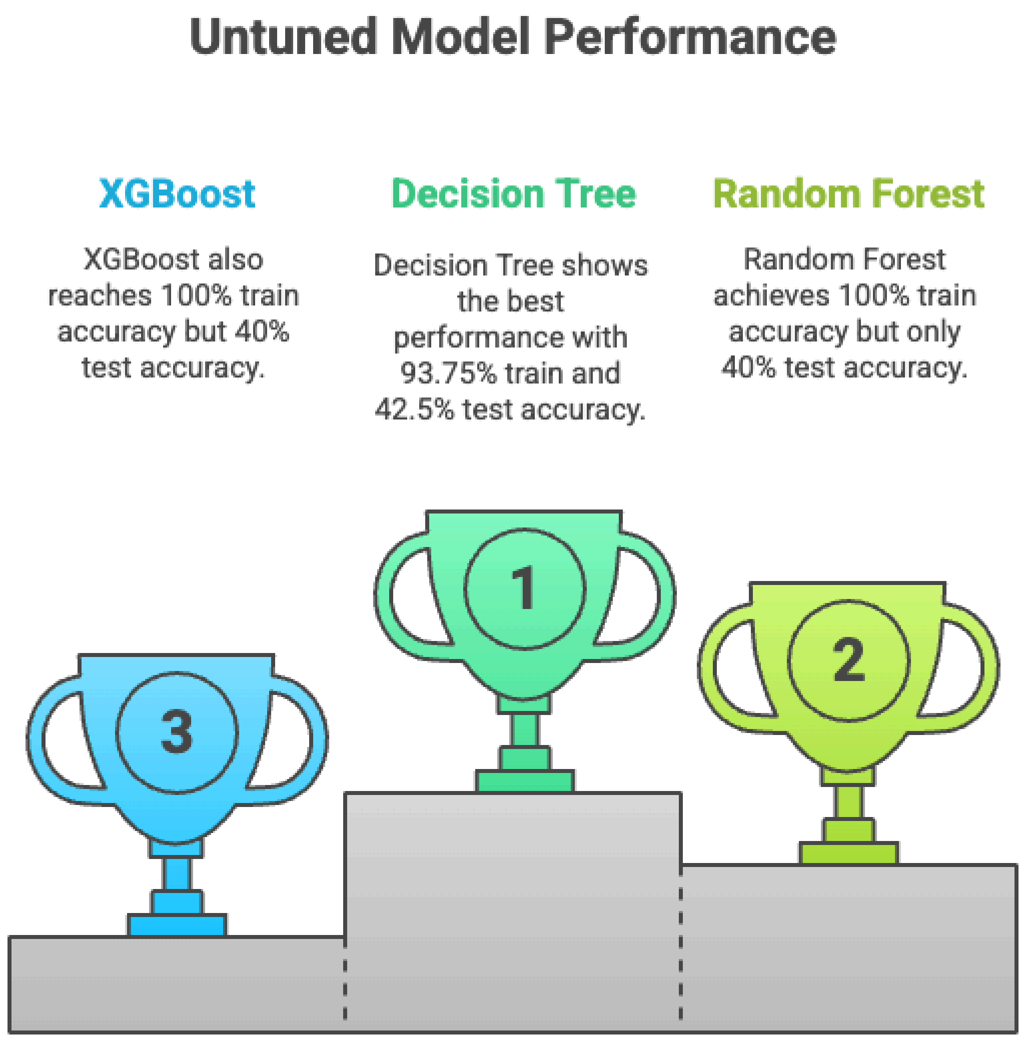
Phase 2: Initial Model Testing

Three Candidate Models

- Decision Tree Classifier
- Random Forest (100 estimators)
- XGBoost Classifier

Why are all models struggling?

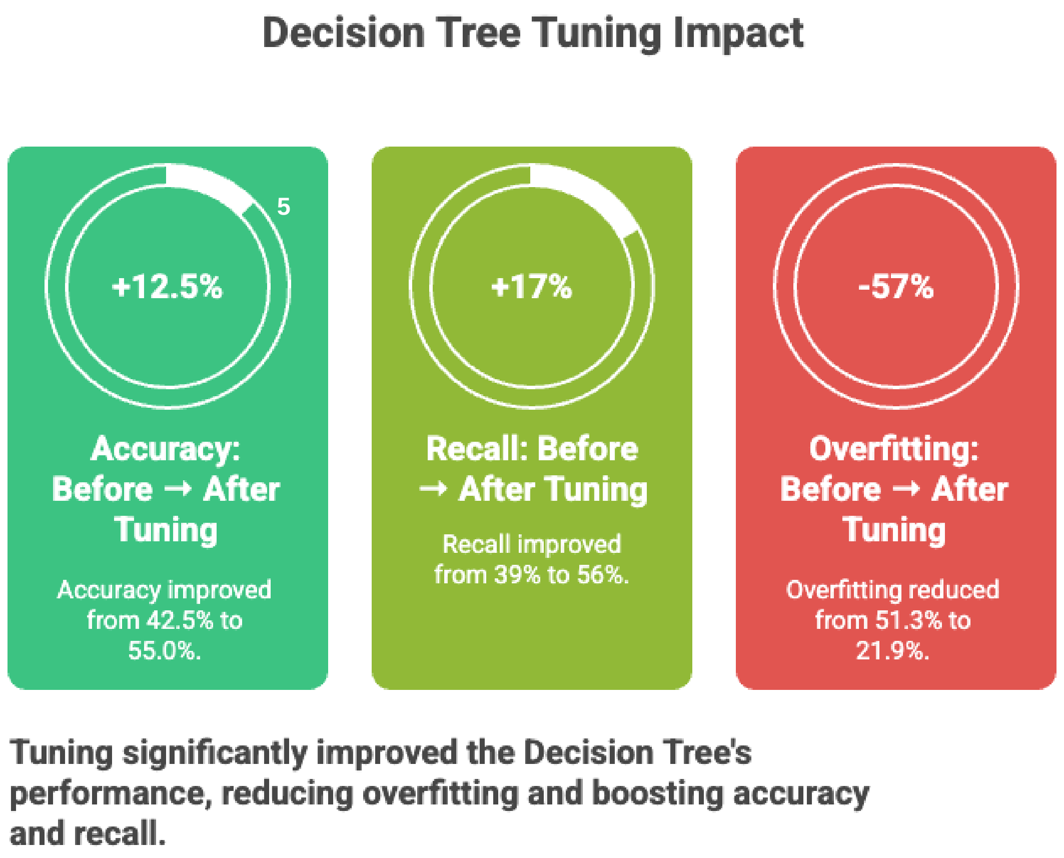
- Dataset size: 160 training samples may be too small for these complex models to generalize
- Feature-to-sample ratio: 15 features with 160 samples gives us only ~10 samples per feature
- Problem difficulty: Severe pollution episodes may be inherently difficult to predict with available features
- Hyperparameters need tuning: Default/conservative settings aren't optimal



Model	Training Accuracy	Testing Accuracy	Overfitting Gap
Decision Tree	93.75%	42.50%	51.25%
Random Forest	100.00%	40.00%	60.00%
XGBoost	100.00%	40.00%	60.00%

Phase 3: Hyperparameter Tuning Results

Model	Test Accuracy	Recall (Severe)	True Positives	False Negatives	Overfitting Gap
Original Decision Tree	42.50%	39%	7	11	51.25%
Tuned Decision Tree ✅	**55.00%**	**56%**	**10**	**8**	**21.88%**
Original Random Forest	40.00%	28%	5	13	60.00%
Tuned Random Forest	37.50%	28%	5	13	53.12%
Original XGBoost	40.00%	33%	6	12	60.00%
Tuned XGBoost	40.00%	33%	6	12	60.00%



Key Takeaways

Decision Tree Success

- +12.5% accuracy improvement
- +17% recall improvement
- -29 pts reduction in overfitting
- Now catches 10/18 severe episodes vs. 7/18 originally

Ensemble Methods Failed

- Random Forest: Tuning made it worse (37.5% accuracy)
- XGBoost: Zero improvement despite 729 combinations tested
- Both models still severely overfit (53-60% gaps)
- originally

Why Simple Won?!

With only 160 training samples, ensemble methods couldn't learn generalizable patterns—they either:

- Over-split data (Random Forest bootstrap sampling)
- Over-boosted noise (XGBoost sequential learning)

Simpler, well-constrained model outperformed complex ensembles

Model Prediction

DEMONSTRATION

Model Prediction & Performance Validation

Our goal was simple:
Can we forecast dangerous pollution days early enough to prevent health risks?

PURPOSE & PIPELINE

1. Load Model: best_model.pkl
2. Load Data: 200 samples
3. Predict: predict() + proba()
4. Evaluate: Accuracy, CM
5. Analyze: City-wise, Errors
6. Summarize: Dashboard

KEY RESULTS

Overall Accuracy: 72.5%

(145 out of 200 correct predictions)

Predicted: Normal 102, Severe 98
Actual: Normal 111 (55.5%), Severe 89 (44.5%)

MODEL LOADED

Tuned Decision Tree

max_depth: 7
min_samples_split: 30

Test Accuracy: 55%
Recall (Severe): 56%

CLASSIFICATION REPORT

	Precision	Recall	F1-Score
Normal Day	77.5%	71.2%	0.742
Severe Episode	67.3%	74.2%	0.706
Macro Avg	72.4%	72.7%	0.724

CONFUSION MATRIX

	Pred Normal	Pred Severe
Actual Normal	TN = 79 (Correct)	FP = 32 (False Alarms)
Actual Severe	FN = 23 (MISSED!)	TP = 66 (Caught)

Key: 74.2% Recall means model catches 74% of severe episodes | 23 severe days missed, 32 false alarms generated

ANALYSIS

Analysis & Summary

Deep-Dive: City Performance, Error Analysis & Feature Importance

- Model achieves 72.5% accuracy with strong recall for severe days (74.2%).
- Balanced predictions verified using classification report and confusion matrix.
- Reliable early-warning capability for air-quality risk detection.

CITY-WISE ACCURACY

City	Samples	Actual	Pred	Acc
Jaipur	28	9	11	78.6%
Lucknow	29	9	14	75.9%
Delhi	29	17	13	72.4%
Chandigarh	29	13	13	72.4%
Amritsar	28	16	18	71.4%
Patna	29	13	16	69.0%
Gurugram	28	12	13	67.9%

ERROR ANALYSIS

FALSE NEGATIVES (Missed Severe)

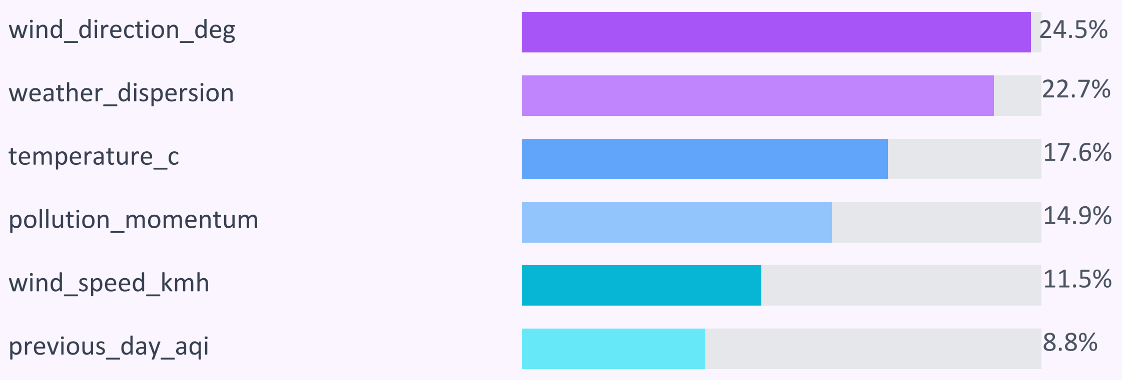
23 severe episodes MISSED!
DANGEROUS - No warning issued

FALSE POSITIVES (False Alarms)

32 normal days flagged as severe
Causes alarm fatigue, wasted resources

FEATURE IMPORTANCE

Top Features Driving Predictions:



FINAL SUMMARY DASHBOARD

MODEL INFORMATION

- Model: Tuned Decision Tree
- Features: 15 predictive features
- Dataset: 200 samples
- Cities: 7 North Indian cities

PREDICTION RESULTS

- Total Predictions: 200
- Correct: 145 (72.5%)
- Predicted Severe: 98
- Best City: Jaipur (78.6%)

CRITICAL METRICS

- True Positives: 66 (caught)
- False Negatives: 23 (missed)
- False Positives: 32 (alarms)
- Recall: 74.2%, Precision: 67.3%

CONCLUSION: Model achieves 72.5% accuracy with 74.2% recall - catches 3 out of 4 severe pollution episodes for public health warnings

IMPACT

Users & Applications

- City-wise insights reveal performance variation and pollution behavior.
- Error analysis highlights missed severe days as the highest-risk cases.
- Weather factors like wind and dispersion drive most prediction outcomes.

Government

CPCB and State Boards can issue early warnings and implement GRAP measures proactively

Healthcare

Hospitals can prepare for respiratory case surges and alert vulnerable patients

Education

Schools can plan activities and make informed decisions about closures

Citizens

Public can plan travel and take protective measures in advance

Practical Applications

- Mobile app early warning alerts
- Automated school closure recommendations
- Traffic management planning
- Industrial activity scheduling

Social Impact

- Reduced health costs from pollution
- Protection for vulnerable populations
- Data-driven policy decisions
- Environmental accountability

NEXT STEPS

Future Improvements

- Integrate satellite, traffic, and emission datasets for richer predictions.
- Upgrade to deep learning and multi-day forecasting models.
- Vision: a scalable pollution-alert system that protects public health.

More Data Sources

◆ Satellite imagery (Sentinel-5P)
◆ Traffic density data
◆ Industrial emission reports
◆ Ground sensor networks

Model Enhancements

◆ Deep learning (LSTM)
◆ Multi-day forecasting
◆ Confidence intervals
◆ Explainable AI (SHAP)

Fairness & Equity

◆ Include smaller cities
◆ Socioeconomic factors
◆ Multi-language support
◆ Bias auditing

Roadmap

Collect 1000+ samples
Phase 1

Real-time API integration
Phase 2

Mobile app deployment
Phase 3

Scale to all India
Phase 4

Predicting Pollution.
Protecting Health.

Thank You