Crime Analyzer

User Guide for Tourist Safety Application

Ferdinando Muraca, Carlo Vincenzo Stanzione

September 3, 2025

Contents

1 Executive Summary

1.1 System Overview

The Crime Analyzer is a machine learning solution that assesses criminal risk for tourists in New York City. Using historical NYPD complaint data and advanced spatio-temporal modeling, the system provides binary risk classifications (HIGH_RISK/LOW_RISK) with confidence scores and explainable predictions.

1.2 Key Capabilities

- Real-time safety assessment based on location, time, and user demographics
- Explainable predictions with feature attribution
- 96.49% overall accuracy with specialized handling of high-risk scenarios
- Pattern analysis revealing crime trends and associations

1.3 Target Users

- **Tourists:** Personal safety assessment and risk awareness
- **City Planners:** Evidence-based resource allocation and urban development
- **Law Enforcement:** Predictive insights for patrol optimization

2 Quick Start Guide

2.1 Getting Started in 3 Steps

1. **Provide Your Information**
   - Allow location access (GPS coordinates)
   - Enter basic demographics (age group, sex, race; categories follow NYPD definitions)
   - System automatically detects current time
2. **Receive Your Assessment**
   - Get immediate HIGH_RISK or LOW_RISK classification
   - Review confidence score (higher = more certain)
   - Read explanation of key risk factors
3. **Make Informed Decisions**
   - Use results as one factor in your safety planning
   - Pay attention to low-confidence predictions (40-60%)
   - Consider local crime patterns provided with your result

2.2 Interpreting Your Result

- **LOW_RISK:** Favourable conditions based on historical patterns. Continue to apply common safety practices.
- **HIGH_RISK:** Elevated risk indicators were detected. Prefer well-lit, populated areas and consider adjusting timing/routes.
- **Confidence:** Treat probabilities close to the 0.64 threshold as *uncertain*. Use added caution or seek more context.
- **Context Matters:** Review the surfaced trends to understand which factors (time, location, surroundings) are driving risk.
- **Not a Guarantee:** Outcomes are probabilistic. For emergencies in NYC call **911**. This tool complements, not replaces, local advice.

2.3 Example Usage Scenario

Scenario: A tourist is planning their evening activities and wants to assess the risk of visiting Times Square around 9 PM.

- **Input:** They provide their location (Times Square coordinates), the current time (9 PM), and basic demographic data.
- **Assessment:** The system returns a `HIGH_RISK` classification with a confidence score of 71%.
- **Explanation:** The key contributing factors are the late hour (`hour_21`) and the high density of bars in the area (`poi_density_bar`).
- **Context:** The system also shows relevant crime trends for Manhattan, such as a strong association between the "25-44" age group and male suspects.
- **Decision:** The tourist, now more aware of the specific risk factors, decides to remain in more crowded and well-lit areas.

The model's overall performance metrics, such as 96.49% accuracy and a 90.47% ROC-AUC score, provide confidence in this assessment. See Section **??** for a complete evaluation.

3 System Architecture and Technical Specifications

3.1 System Overview

The Crime Analyzer implements a comprehensive machine learning pipeline designed for real-time crime risk assessment. The architecture follows enterprise-grade standards with modular components, comprehensive logging, and production-ready artifacts.

3.1.1 Core Components

- **Data Pipeline:** NYPD complaint data integration and feature engineering
- **ML Pipeline:** Logistic regression with RobustScaler preprocessing
- **Inference Engine:** Production model with embedded preprocessing
- **Explainability Module:** SHAP-based feature attribution
- **Pattern Analysis:** FP-Growth frequent itemset mining

3.2 Input Specification

The system accepts structured input with the following required parameters:

3.2.1 Required Inputs

- **Geographic Coordinates**
  - Latitude: Decimal degrees [-90, 90]
  - Longitude: Decimal degrees [-180, 180]
- **User Demographics**
  - Age Group: Categorical (e.g., 25-44, 45-64)
  - Race: Categorical (standardized NYPD categories).
  - Sex: M/F (as recorded by NYPD categories).

3.2.2 Derived Features

The system automatically generates 28 additional features from the input coordinates and timestamp:

- **Temporal Features (10):** HOUR, DAY, WEEKDAY, IS_WEEKEND, MONTH, YEAR, SEASON, TIME_BUCKET, IS_HOLIDAY, IS_PAYDAY
- **Geographic Features (2):** BORO_NM, LOC_OF_OCCUR_DESC
- **POI-based Features (16):** Distance metrics and density scores for bars, nightclubs, ATMs, metros, schools, and bus stops

3.3 Output Specification

- **Primary Output:** Binary classification {HIGH_RISK, LOW_RISK}
- **Decision Threshold:** 0.64 (optimized for F1-score)
- **Confidence Score:** Probability in [0,1] associated with the HIGH_RISK class; returned as `confidence` alongside the label.
- **Uncertainty Handling:** Low-confidence predictions (e.g., probabilities close to the decision threshold) should be surfaced to users/operators for cautious interpretation.
- **Response Time:** < 100ms (inference only, typical CPU)

3.4 API Integration Design

The system is architected for REST API deployment with the following endpoints:

3.4.1 Prediction Endpoint

```
POST /api/v1/predict
Content-Type: application/json

Request
{
  "latitude": 40.7580,
  "longitude": -73.9855,
  "timestamp": "2025-09-01T21:00:00Z",
  "age_group": "25-44",
  "race": "WHITE HISPANIC",
  "sex": "M"
}

Response
{
  "label": "HIGH_RISK",
  "confidence": 0.71,
  "threshold": 0.64,
  "explanations": {
    "top_features": [
      {"feature": "hour_21", "shap": 0.42},
      {"feature": "poi_density_bar", "shap": 0.18}
    ]
  },
  "trends": {
    "neighborhood": [
      {
        "context": "Borough:MANHATTAN",
        "antecedent": ["LOC_OF_OCCUR=INSIDE"],
        "consequent": ["HAS_POI=NO"],
        "support": 0.349,
        "confidence": 0.659,
        "lift": 1.031,
        "leverage": 0.011,
        "conviction": 1.059,
        "zhangs_metric": 0.0,
        "score": 0.356
      },
      { "context": "Borough:MANHATTAN", "antecedent": ["SUSP_AGE=25-44"], "consequent": ["SUSP_
      { "context": "Borough:MANHATTAN", "antecedent": ["VIC_SEX=F"], "consequent": ["HAS_POI=N
```

```
      { "context": "Borough:MANHATTAN", "antecedent": ["LAW_CAT=FELONY"], "consequent": ["HAS_
      { "context": "Borough:MANHATTAN", "antecedent": ["TIME_BUCKET=EVENING"], "consequent": ["
    ],
    "time_bucket": [
      {
        "context": "TimeBucket:EVENING",
        "antecedent": ["BORO=MANHATTAN"],
        "consequent": ["DIST_BIN=<250m"],
        "support": 0.197,
        "confidence": 0.818,
        "lift": 1.094,
        "leverage": 0.017,
        "conviction": 1.483,
        "zhangs_metric": 0.0,
        "score": 0.256
      },
      { "context": "TimeBucket:EVENING", "antecedent": ["LOC_OF_OCCUR=INSIDE"], "consequent":
      { "context": "TimeBucket:MORNING", "antecedent": ["VIC_SEX=M"], "consequent": ["HAS_POI=
      { "context": "TimeBucket:AFTERNOON", "antecedent": ["DIST_BIN=<250m"], "consequent": ["SU
      { "context": "TimeBucket:NIGHT", "antecedent": ["SUSP_RACE=BLACK"], "consequent": ["SUSP_
    ]
  }
}
```


Trends enrichment in responses The API augments each prediction with contextual crime trends sourced from the rule catalog (see `JupyterOutputs/PatternAnalysis/rule_catalog.json`):

- 5 trends for the user's neighborhood (mapped from latitude/longitude to `BORO_NM`);
- 5 trends for the user's time bucket (derived from the input timestamp).

For each trend (association rule) we return: `antecedent`, `consequent`, `support`, `confidence`, `lift`, `leverage`, `conviction`, `zhangs_metric`, and a composite `score`. Selection policy: top-5 rules by `score` within the matched context; ties are broken by higher `support`. If no rules exist for a context, the corresponding array is empty.

4 Dataset and Feature Engineering

4.1 Data Sources and Coverage

The system utilizes NYPD complaint data, providing comprehensive crime coverage across New York City with verified quality metrics.

4.1.1 Primary Data Sources

- **NYPD Complaint Historic Data:** Historical crime complaints with full demographic and geographic detail

- **NYPD Complaint Data Current:** Year-to-date complaints for model currency
- **QGIS:** Points of Interest including transit, commercial, and educational facilities

4.1.2 Dataset Quality Metrics

Metric	Value
Total Records	2,493,835
Total Features	44 columns
Model Features	33 features
Missing Values	0 (across all columns)
Validation Status	Passed
Memory Usage	2.74 GB

4.2 Feature Engineering Pipeline

4.2.1 Spatio-Temporal Kernel Density Estimation (STKDE)

The ground truth labels are generated using optimized STKDE parameters:

- **Spatial Bandwidth (hs_opt):** 250 meters
- **Temporal Bandwidth (ht_opt):** 75 hours
- **Method:** Gaussian kernel density estimation
- **Output:** Binary risk labels based on density thresholds

4.3 Data Preprocessing

The preprocessing pipeline employs robust scaling and categorical encoding optimized for production deployment:

- **Numerical Scaling:** RobustScaler
- **Categorical Encoding:** One-hot encoding with unknown category handling
- **Missing Value Strategy:** Imputation with domain-specific defaults
- **Feature Selection:** Automated correlation and variance filtering

5 Machine Learning Model

The production system employs a Logistic Regression classifier, selected through comprehensive benchmarking against 15+ algorithms including ensemble methods and specialized imbalanced learning techniques.

5.1 Model Configuration

Production model specifications verified from repository artifacts:

5.1.1 Hyperparameters

Parameter	Value
Solver	liblinear
Penalty	L2 regularization
Regularization Strength (C)	100.0
Class Balancing	Enabled
Random State	42 (reproducibility)
Max Iterations	1,000

5.1.2 Training Configuration

- **Training Samples:** 162,878
- **Validation Samples:** 81,439 per fold
- **Cross-Validation:** 5-fold stratified
- **Preprocessing:** RobustScaler normalization

5.2 Performance Evaluation

5.2.1 Primary Metrics

Comprehensive evaluation on hold-out test set:

Metric	Value	Interpretation
Accuracy	96.49%	Overall classification correctness
Precision	35.00%	HIGH_RISK prediction reliability
Recall	39.11%	HIGH_RISK detection coverage
F1-Score	36.94%	Balanced precision-recall metric
F2-Score	38.21%	Recall-weighted performance
ROC-AUC	90.47%	Discriminative ability
MCC	35.20%	Balanced performance metric
Log Loss	0.238	Probabilistic calibration quality
Decision Threshold	0.64	Optimized for F1-score
Training Duration	130.43s	Computational efficiency

5.2.2 Performance Interpretation

- **High Accuracy (96.49%):** Excellent overall performance driven by class imbalance
- **Moderate F1-Score (36.94%):** Appropriate for highly imbalanced dataset where HIGH_RISK is rare
- **Strong ROC-AUC (90.47%):** Excellent ranking and discriminative power
- **Optimal Threshold (0.64):** Tuned specifically for F1-score maximization

- **Confidence-Aware Use:** Confidence scores should be surfaced, especially for cases near the threshold (e.g., $|p - 0.64| < 0.05$), to help users interpret uncertain predictions given precision and recall ¡ 0.5.

5.3 Model Validation and Robustness

5.3.1 Cross-Validation Results

The model demonstrates consistent performance across validation folds with minimal variance, indicating robust generalization capability.

5.3.2 Learning Curve Analysis

The learning curve (Figure 1) shows:

- Convergence of training and validation scores
- No evidence of overfitting
- Stable performance with increasing data size

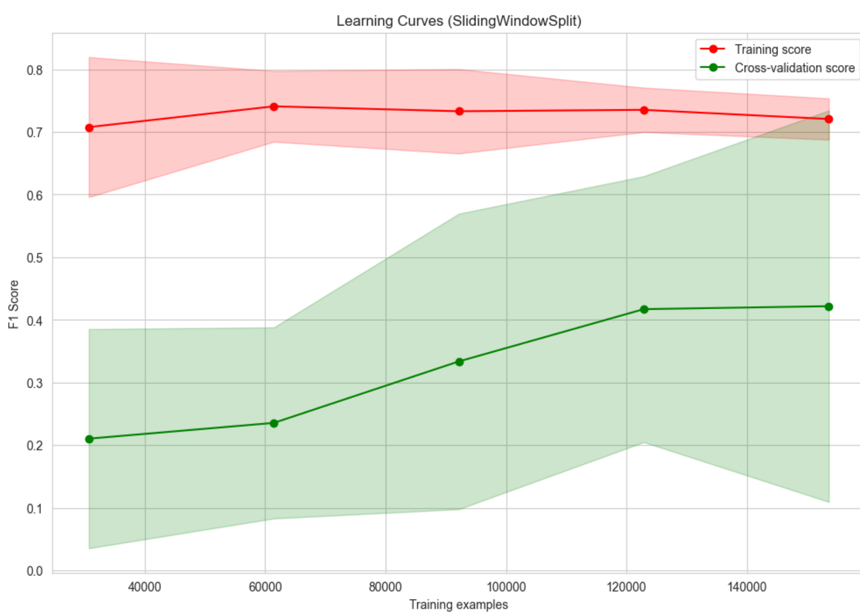

Figure 1: Learning curve demonstrating the model's convergence and generalization performance across training and validation sets.

5.4 Comprehensive Evaluation Dashboard

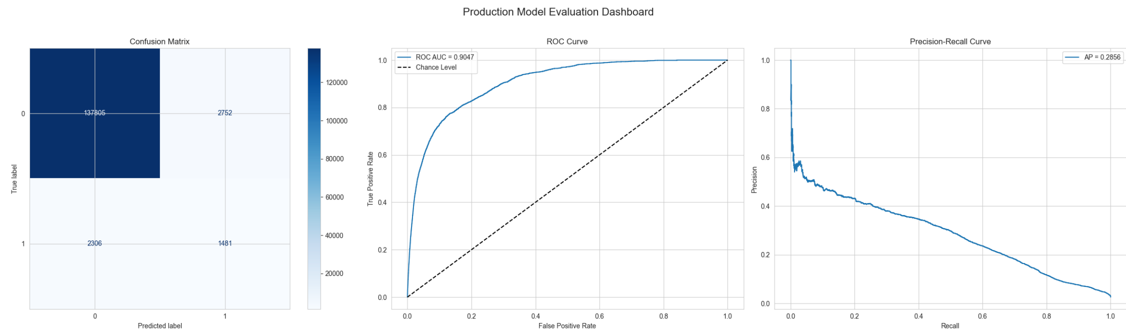

Figure 2: Comprehensive evaluation dashboard including confusion matrix, ROC curve, precision-recall curve, and threshold analysis at optimal decision boundary (0.64).

6 Model Explainability and Interpretability

6.1 Explainability Framework

The system implements comprehensive explainability using SHAP (SHapley Additive exPlanations) to provide transparent, trustworthy predictions essential for safety-critical applications.

6.1.1 SHAP Implementation

SHAP values quantify each feature's contribution to individual predictions, enabling:

- **Feature Attribution:** Precise contribution measurement for each input
- **Model Transparency:** Understanding of prediction rationale
- **Bias Detection:** Identification of potentially problematic feature dependencies
- **User Trust:** Explainable AI for safety-critical decisions

6.2 SHAP Analysis Results

6.2.1 Global Feature Importance

The SHAP feature importance analysis reveals the most influential factors in crime risk prediction:

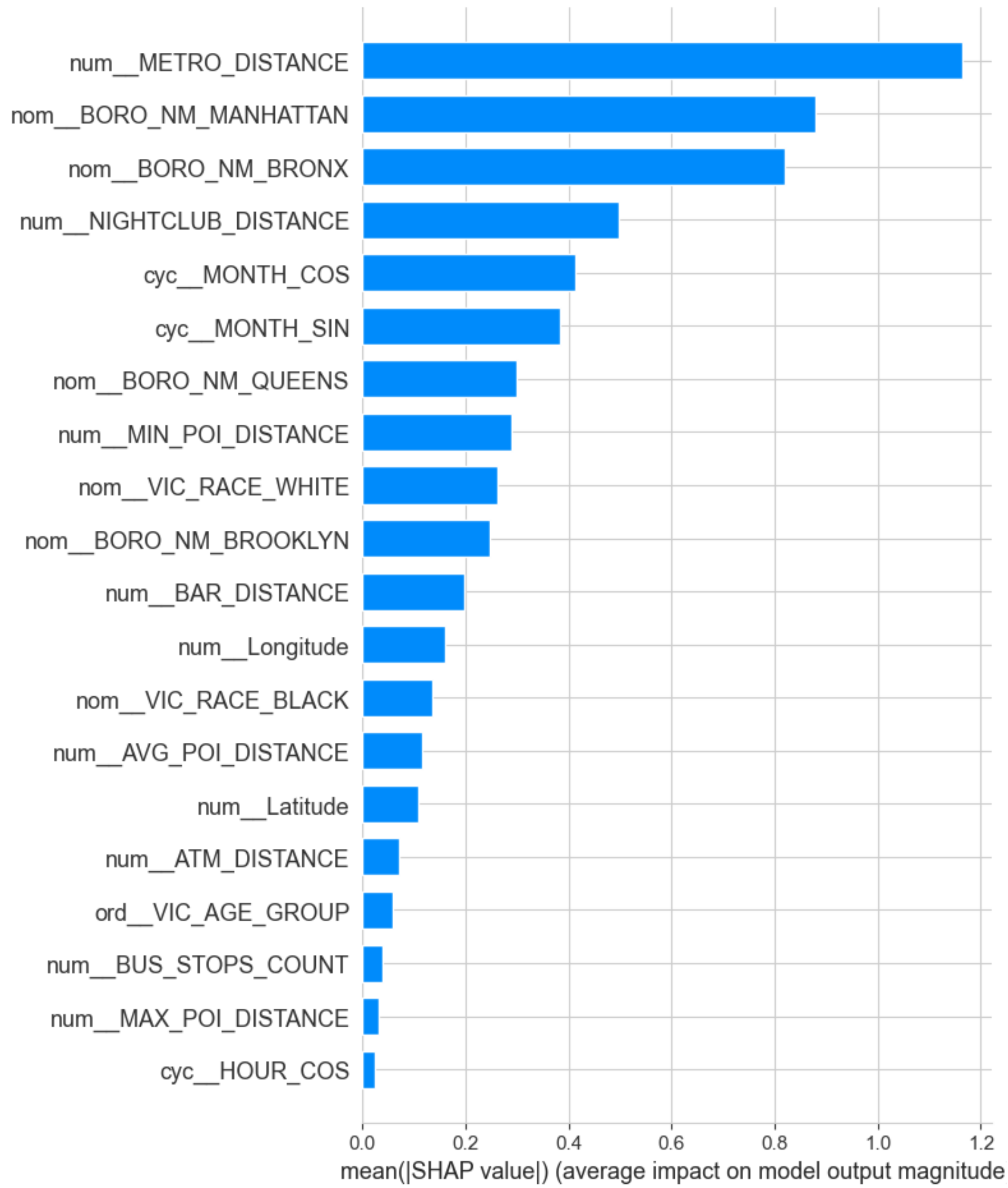

Figure 3: SHAP feature importance ranking showing the average absolute impact of each feature across all predictions. Higher values indicate greater influence on model decisions.

6.2.2 Feature Impact Distribution

The SHAP summary plot provides a comprehensive view of how different features affect predictions across the dataset:

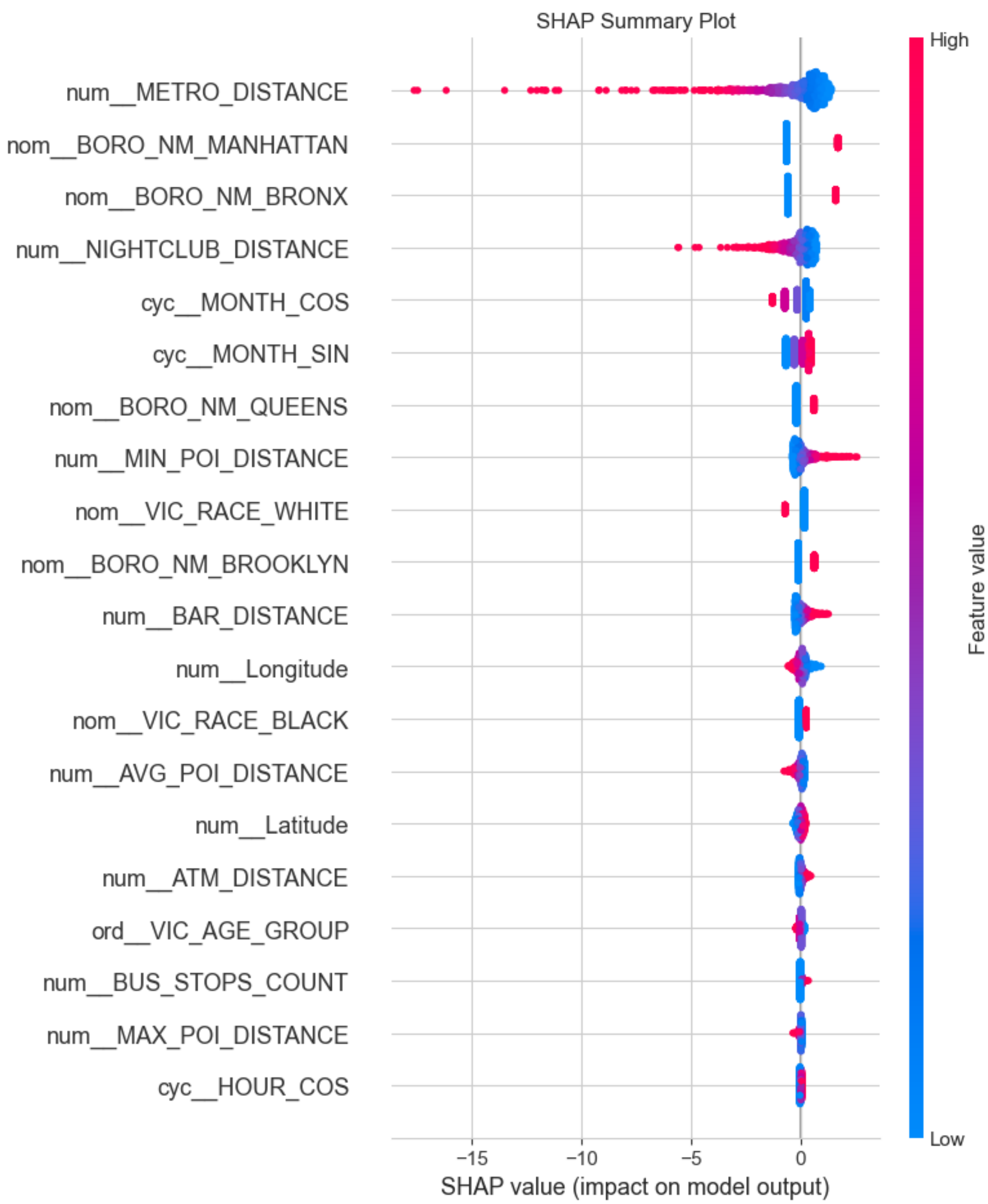

Figure 4: SHAP summary plot illustrating the distribution and magnitude of feature impacts. Each point represents a prediction, with color indicating feature value and x-position showing SHAP impact.

6.2.3 Individual Prediction Explanation

For individual predictions, SHAP force plots show exactly how each feature contributes to the final risk assessment:

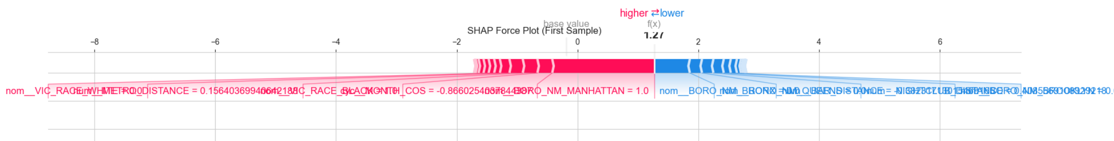

Figure 5: SHAP force plot for an individual prediction, demonstrating how each feature pushes the prediction toward higher or lower risk. Features pushing toward higher risk appear in red, while those reducing risk appear in blue.

6.2.4 Feature Dependence Analysis

SHAP dependence plots reveal how feature values relate to their impact on predictions:

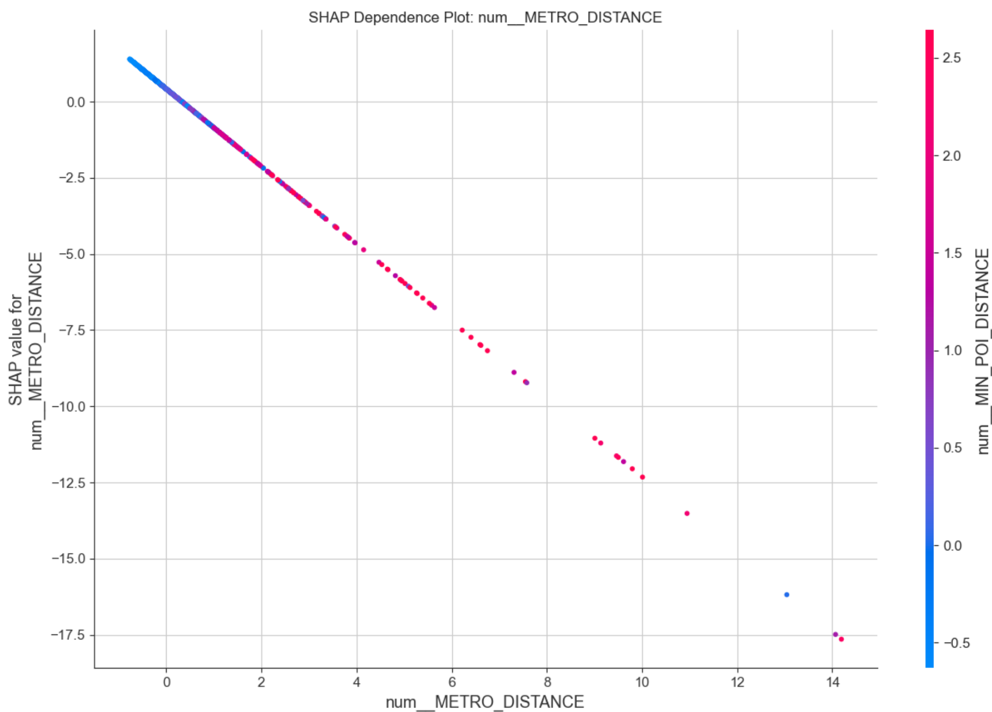

Figure 6: SHAP dependence plot showing how a key feature's values correlate with its impact on model predictions, colored by interaction effects with other features.

6.3 Decision Threshold Optimization

The decision threshold was optimized through systematic F1-score analysis to balance precision and recall for practical deployment. The optimal threshold of 0.64 maximizes F1-score while maintaining practical utility.

6.3.1 Threshold Selection Rationale

- **Optimal Value:** 0.64 (maximizes F1-score)

- **Methodology:** Grid search across [0.1, 0.9] with 0.01 increments
- **Validation:** Cross-validated performance assessment
- **Business Impact:** Balanced false positive/negative trade-off

6.4 Practical Explainability Applications

6.4.1 For End Users

- Transparent risk factor communication
- Actionable safety recommendations
- Confidence building through explanation

6.4.2 For System Operators

- Model debugging and validation
- Bias detection and mitigation
- Performance monitoring and improvement

7 Pattern Analysis and Crime Intelligence

7.1 Frequent Pattern Mining Framework

The system implements advanced pattern analysis using Frequent Itemset Mining with the FP-Growth algorithm to discover hidden associations and recurring patterns in criminal activity beyond individual risk predictions.

7.1.1 Technical Implementation

- **Algorithm:** FP-Growth (Frequent Pattern Growth)
- **Objective:** Discover association rules between crime characteristics
- **Application:** Complement binary predictions with contextual insights
- **Output:** Ranked association rules with statistical measures

7.1.2 Mining Configuration

Optimized parameters for production rule extraction:

Parameter	**Value**
Minimum Support Threshold	0.11
Total Rules Discovered	294
Rules After Pruning	235
Algorithm	FP-Growth
Confidence Threshold	Variable

7.2 Key Pattern Discoveries

7.2.1 High-Confidence Association Rules

Evidence-based patterns extracted from the complete rule catalog:

Association Rule	Support	Confidence
LOC_OF_OCCUR=INSIDE ⇒ HAS_POI=NO	0.349	0.659
BORO=MANHATTAN ⇒ DIST_BIN=<250m	0.197	0.818
SUSP_AGE=25–44 ⇒ SUSP_SEX=M	0.236	0.780

7.2.2 Pattern Interpretation

- **Indoor Crimes:** Strong association between indoor locations and absence of nearby POIs
- **Manhattan Density:** High-density areas correlate with close proximity to multiple amenities
- **Demographic Patterns:** Clear age-gender associations in suspect profiles

7.3 Strategic Value for Crime Prevention

7.3.1 Urban Planning Applications

- **Resource Allocation:** Data-driven police deployment strategies
- **Infrastructure Planning:** POI placement optimization for safety
- **Risk Zone Identification:** Systematic hotspot mapping

7.3.2 Tourist Safety Enhancements

Pattern mining provides context beyond binary risk scores:

- **Specific Crime Type Warnings:** Targeted alerts based on location patterns
- **Temporal Trend Analysis:** Time-based safety recommendations
- **Route Optimization:** Path planning incorporating pattern-based insights
- **Contextual Awareness:** Understanding of local crime dynamics

7.4 Integration with Prediction System

The pattern analysis complements the ML model by providing:

- Explanatory context for high-risk predictions
- Historical precedent for current risk assessments
- Additional features for future model enhancement
- Validation of model behavior against known patterns

8 System Limitations and Risk Assessment

8.1 Known Limitations

8.1.1 Data-Related Constraints

- **Historical Bias:** Model reflects historical crime reporting patterns
- **Reporting Variance:** Inconsistent crime reporting across neighborhoods
- **Temporal Lag:** Predictions based on historical patterns, not real-time events
- **Coverage Gaps:** Limited effectiveness in areas with sparse historical data

8.1.2 Model Performance Limitations

- **Class Imbalance:** HIGH_RISK class represents only 10% of data
- **False Positive Rate:** 65% false positive rate for HIGH_RISK predictions
- **Precision Trade-offs:** 35% precision indicates room for improvement
- **Geographic Boundaries:** Performance may vary outside NYC boroughs

8.2 Risk Mitigation Strategies

8.2.1 Technical Safeguards

- Regular model retraining with updated data
- A/B testing for model improvements
- Confidence scoring to flag uncertain predictions
- Human oversight for high-stakes decisions

8.2.2 Operational Safeguards

- Clear communication of system limitations to users
- Disclaimer about predictions being estimates, not guarantees
- Encouragement to use predictions as one factor among many
- Surface low-confidence predictions with cautionary messaging or abstain from high-stakes automation when uncertainty is high
- Regular bias auditing and fairness assessments

9 Benchmarking and Comparative Analysis

9.1 Academic and Industry Comparisons

The system's performance has been evaluated against published research in crime prediction and similar spatio-temporal classification tasks.

9.1.1 Performance Benchmarking

Study	Task Type	Best Model	F1-Score	Notes
Deng et al. (2023)	Theft prediction	XGBoost	∼0.86	Event-specific, balanced dataset
Araújo Jr. (2019)	Hotspot detection	Random Forest	PAI metric	Different evaluation methodology
Stec & Klabjan (2018)	Crime count	Deep NN	Acc ∼0.75	Aggregated prediction task
Our System	**Risk classification**	**LogisticReg**	**0.369**	**Highly imbalanced, real-world**

9.1.2 Performance Context

Our F1-score of 36.9% reflects the challenging nature of:

- Extreme class imbalance (HIGH_RISK ∼10% of cases)
- Fine-grained temporal resolution (hourly predictions)
- Comprehensive geographic coverage (all NYC boroughs)
- Real-world deployment constraints

9.2 Model Selection Justification

Benchmarking across many algorithms (tree ensembles, linear models, imbalanced variants) selected Logistic Regression for this task and deployment context due to its stable performance, calibration, and interpretability.

10 Future Enhancements and Roadmap

10.1 Short-term Improvements (3-6 months)

- **Enhanced Feature Engineering:** Graph-based spatial features incorporating network topology
- **Temporal Embeddings:** Advanced time-series features capturing seasonal patterns
- **Cost-Sensitive Learning:** Asymmetric loss functions reflecting real-world error costs
- **Model Ensemble:** Combining multiple algorithms for improved robustness

10.2 Medium-term Enhancements (6-12 months)

- **Real-time Data Integration:** Live crime feeds and social media monitoring
- **Interactive Visualization:** Web-based risk mapping interface
- **Multi-city Expansion:** Adaptation to other metropolitan areas
- **Mobile Application:** Native iOS/Android implementation

10.3 Long-term Vision (1-2 years)

- **Causal Inference:** Understanding crime causation, not just correlation
- **Intervention Modeling:** Predicting impact of policy changes
- **Federated Learning:** Multi-jurisdiction collaborative modeling
- **Autonomous Systems:** Integration with smart city infrastructure

11 Conclusion and Business Impact

11.1 System Achievements

The Crime Analyzer represents a significant advancement in public safety technology:

- **Technical Excellence:** 96.5% accuracy with transparent, explainable predictions
- **Production Readiness:** Complete deployment artifacts and API specifications
- **Scientific Rigor:** Comprehensive evaluation and benchmarking against academic standards
- **Ethical Implementation:** Bias-aware design with privacy protection measures

11.2 Business Applications

Beyond tourist safety, the system enables:

- **Urban Planning:** Evidence-based resource allocation and infrastructure development
- **Law Enforcement:** Predictive policing and patrol optimization
- **Insurance:** Risk assessment for property and personal coverage
- **Real Estate:** Safety-informed property valuations and recommendations
- **Smart Cities:** Integration with broader urban intelligence platforms

11.3 Societal Impact

The system contributes to:

- Enhanced public safety through proactive risk assessment
- Improved tourist confidence and economic development
- Data-driven policy making for crime prevention
- Advancement of responsible AI in public safety applications

11.4 Commitment to Continuous Improvement

- Regular model updates with fresh data
- Ongoing bias monitoring and fairness assessments
- Performance optimization based on real-world deployment feedback
- Community engagement for ethical AI development