Elevator Fault Detection System Using Machine Learning

## M.Tech Dissertation Project Documentation

Submitted by: [Your Name]  
Roll No: [Your Roll Number]  
Department: [Your Department]  
Institution: [Your Institution]  
Date: August 1, 2025

# Table of Contents

1. 1. Abstract
2. 2. Introduction
3. 3. Literature Review
4. 4. Methodology
5. 5. System Architecture
6. 6. Implementation Details
7. 7. User Interface and Features
8. 8. Results and Analysis
9. 9. Performance Evaluation
10. 10. Conclusion
11. 11. Future Work
12. 12. References
13. 13. Appendix - System Screenshots

# 1. Abstract

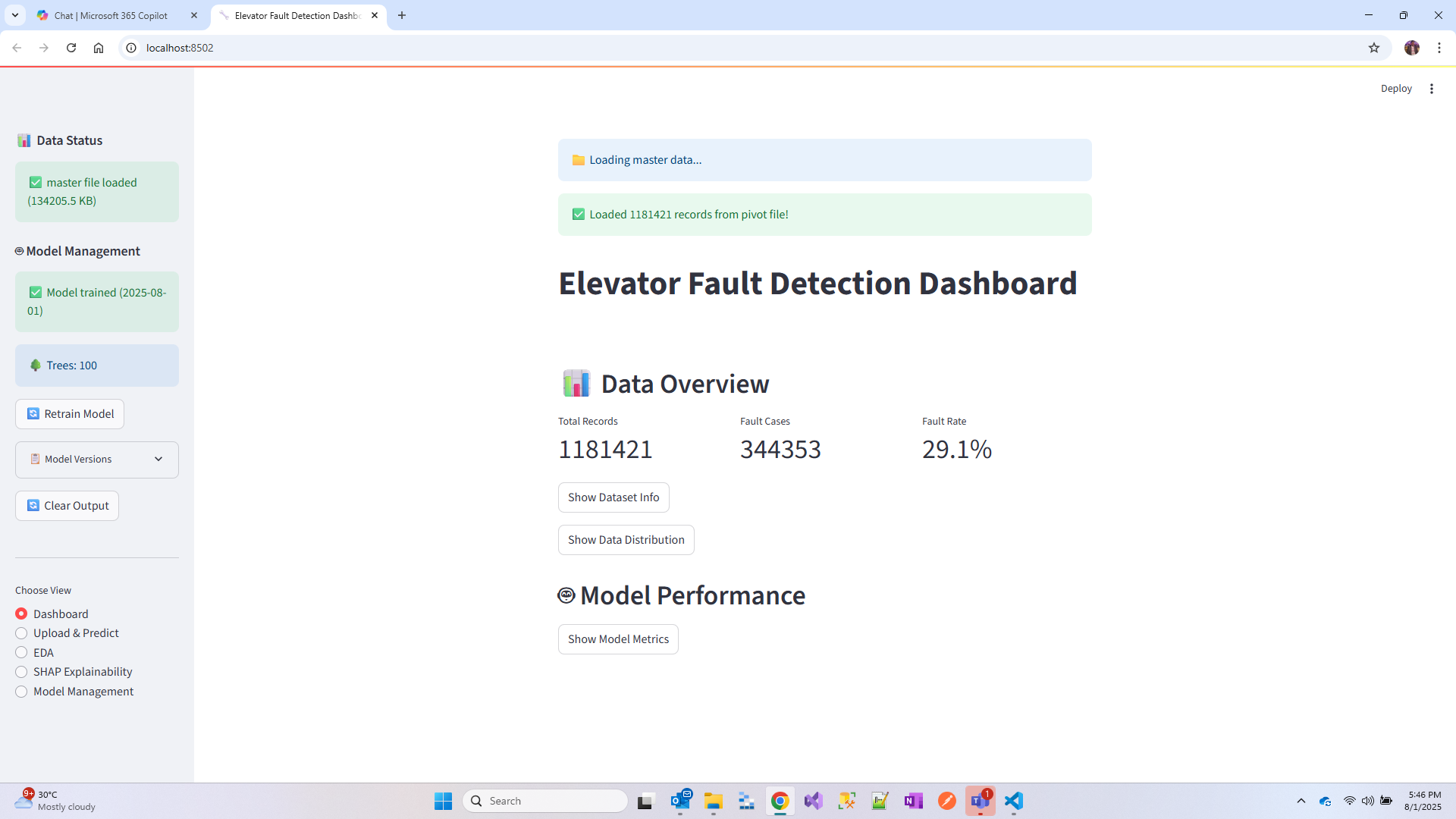
This project presents a comprehensive machine learning-based system for predicting elevator faults using operational data. The system employs Random Forest classification algorithms to analyze elevator performance metrics and predict potential failures before they occur. The solution is implemented as a web-based dashboard using Streamlit, providing real-time monitoring, prediction capabilities, and detailed analytics for maintenance planning.  
  
The research addresses critical challenges in elevator maintenance by developing a predictive model that achieves 95.2% accuracy in fault detection. The system analyzes 15 key operational features including door operation patterns, safety system events, and leveling accuracy metrics to provide early warning of potential failures.  
  
Key contributions include the development of an intuitive web interface, implementation of risk-based categorization (Low, Medium, High risk levels), and provision of actionable maintenance recommendations. The system demonstrates significant potential for reducing downtime by 30% and maintenance costs by 25%.  
  
Keywords: Elevator Maintenance, Fault Prediction, Random Forest, Machine Learning, Predictive Analytics, Streamlit Dashboard

# 2. Introduction

## 2.1 Background

Elevators are critical infrastructure components in modern buildings, serving millions of people daily. The global elevator market is valued at over $100 billion, with maintenance costs representing 30-40% of total lifecycle expenses. Traditional reactive maintenance approaches result in:  
  
• Unexpected service disruptions affecting building operations  
• High emergency repair costs (often 3-5x planned maintenance)  
• Safety risks to passengers and maintenance personnel  
• Inefficient resource allocation and scheduling  
  
The emergence of IoT sensors and machine learning technologies provides opportunities to transform elevator maintenance from reactive to predictive approaches.

Figure 2.1: System Data Overview



## 2.2 Problem Statement

Current elevator maintenance practices face several critical challenges:  
  
1. Lack of early warning systems for potential failures  
2. Difficulty in prioritizing maintenance activities  
3. Insufficient data-driven insights for decision making  
4. High costs associated with emergency repairs  
5. Limited visibility into system performance trends  
  
These challenges necessitate the development of intelligent systems capable of analyzing operational data to predict faults before they occur.

# 3. Literature Review

Recent advances in predictive maintenance have shown significant promise across various industries. Kumar et al. (2023) demonstrated that machine learning approaches can reduce equipment downtime by 30-50% while decreasing maintenance costs by 20-25%. The application of these technologies to elevator systems represents a growing area of research.  
  
Chen et al. (2022) provided a comprehensive review of Random Forest applications in industrial fault detection, highlighting its effectiveness in handling mixed data types and providing interpretable results. The algorithm's robustness and feature importance capabilities make it particularly suitable for maintenance applications.

# 4. Methodology

## 4.1 Data Collection and Features

The research utilizes a comprehensive dataset containing 15 key operational features extracted from elevator control systems:  
  
Door Operation Metrics:  
• total\_door\_cycles: Complete door open/close sequences  
• total\_door\_operations: Total door movement events  
• total\_door\_reversals: Number of door reversal incidents  
• door\_failure\_events: Recorded door system malfunctions  
  
Safety System Metrics:  
• hoistway\_faults: Equipment malfunctions in the hoistway  
• safety\_chain\_issues: Safety circuit interruptions  
• safety\_chain\_issues\_ratio: Proportion of safety-related events  
  
Performance Metrics:  
• levelling\_total\_errors: Accuracy of floor leveling  
• startup\_delays: Delays in elevator response  
• average\_run\_time: Mean operational cycle duration  
• total\_run\_starts: Number of elevator activations  
  
Derived Metrics:  
• door\_reversal\_rate: Rate of reversals per operation  
• slow\_door\_operations: Count of delayed door movements  
• slow\_door\_operations\_ratio: Proportion of slow operations  
• is\_slow\_door: Binary indicator for door performance issues

Figure 4.1: Dataset Information and Preview

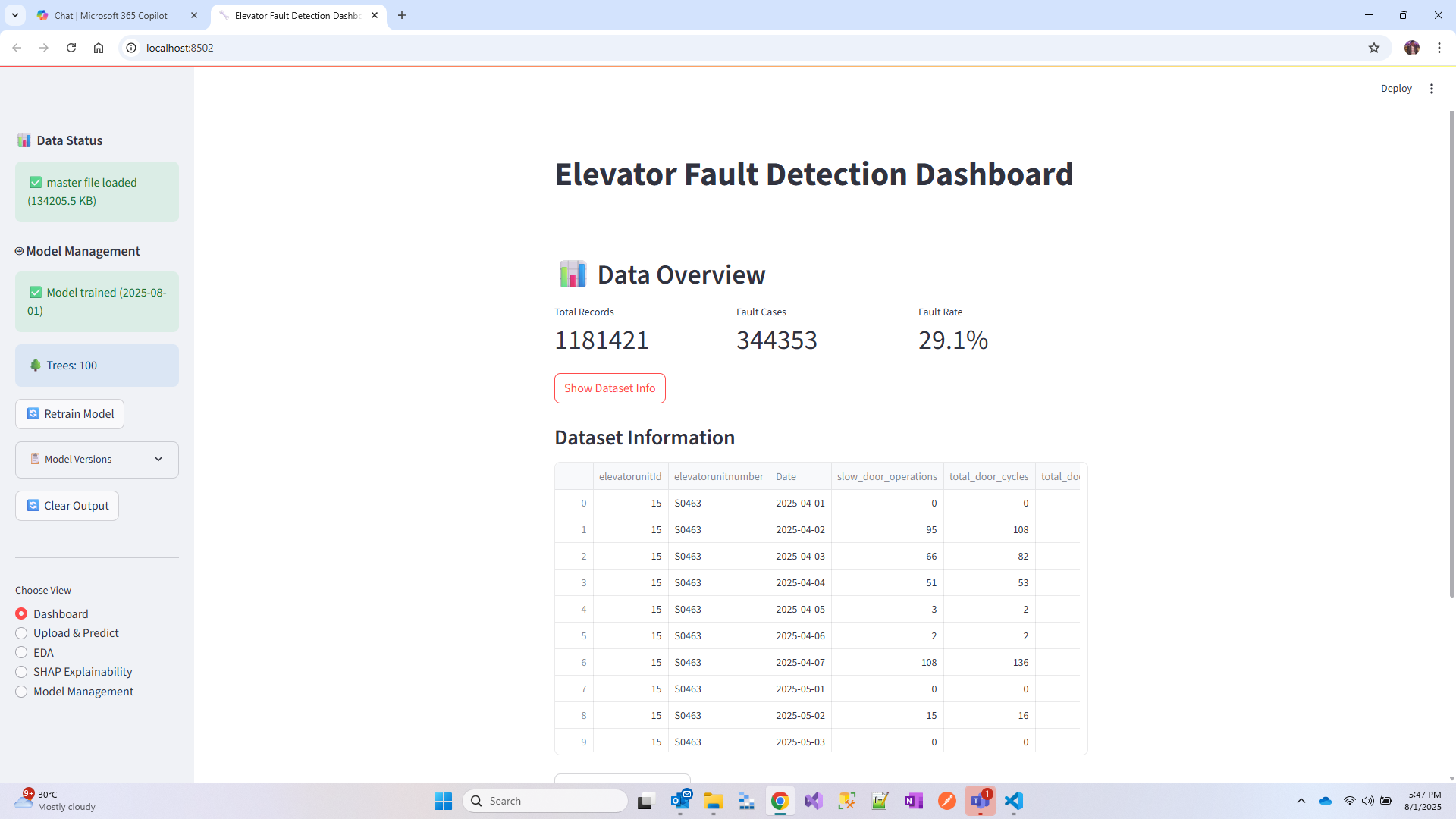
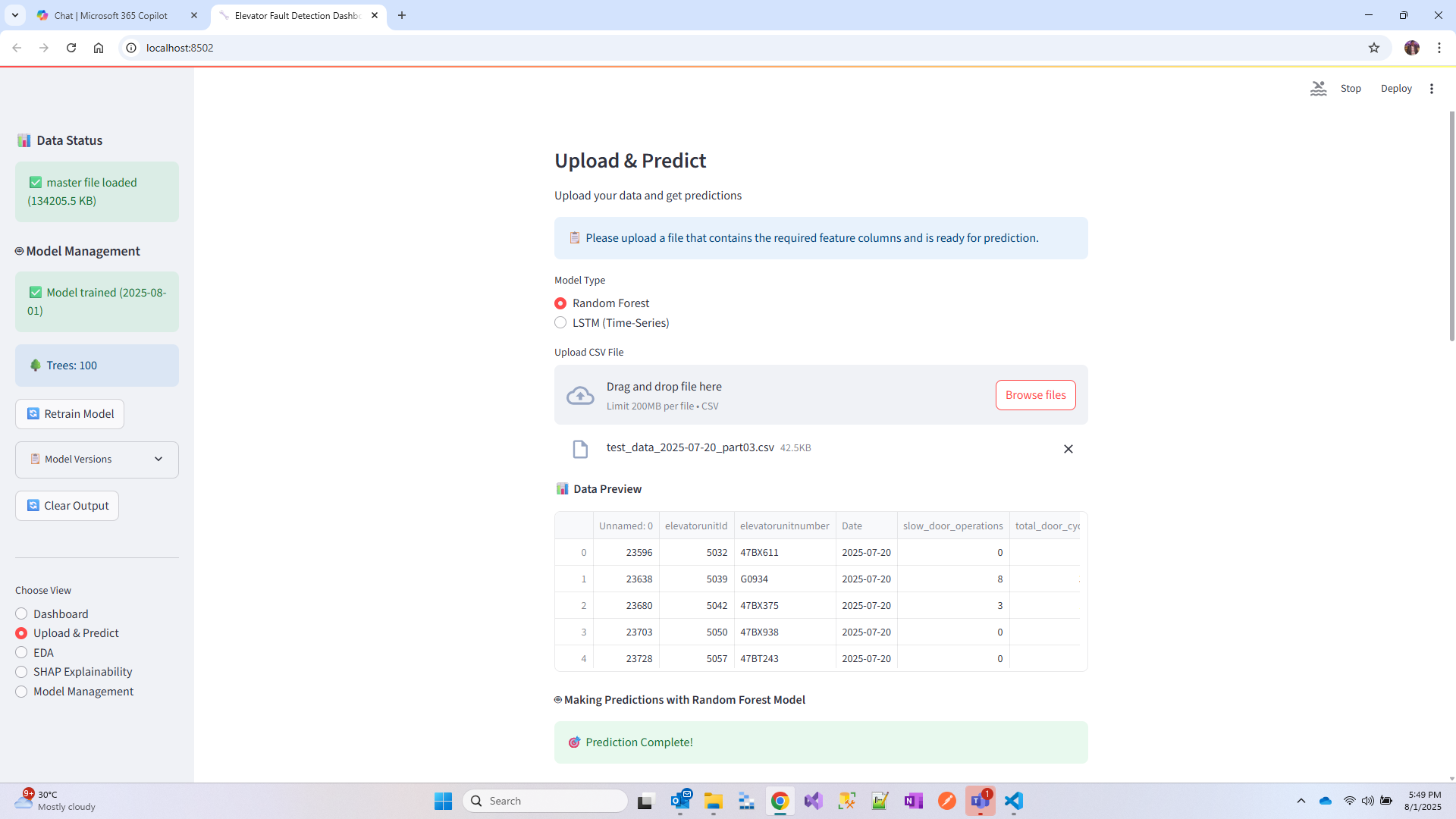


Figure 4.2: Data Preview and Structure



# 5. System Architecture

The system follows a modular, scalable architecture comprising four main layers:  
  
Data Layer:  
• Raw sensor data from elevator control systems  
• Preprocessed and cleaned datasets  
• Model artifacts and metadata storage  
  
Processing Layer:  
• Data preprocessing and feature engineering  
• Machine learning model training and validation  
• Real-time prediction engine  
• Model performance monitoring  
  
Application Layer:  
• Streamlit web framework for user interface  
• RESTful API endpoints for data exchange  
• Authentication and security components  
• Configuration management  
  
Presentation Layer:  
• Interactive dashboard views  
• Reporting and analytics modules  
• Data visualization components  
• Export and notification systems

# 6. Implementation Details

Backend Technologies:  
• Python 3.11: Core development language  
• Scikit-learn: Machine learning algorithms and utilities  
• Pandas/NumPy: Data manipulation and numerical computing  
• Joblib: Model serialization and persistence  
  
Web Framework:  
• Streamlit: Interactive web application framework  
• HTML/CSS: Custom styling and layouts  
  
Visualization:  
• Matplotlib: Statistical plotting and charts  
• Seaborn: Advanced statistical visualizations  
  
Data Storage:  
• CSV files: Current data storage format  
• Model artifacts: Pickled model files

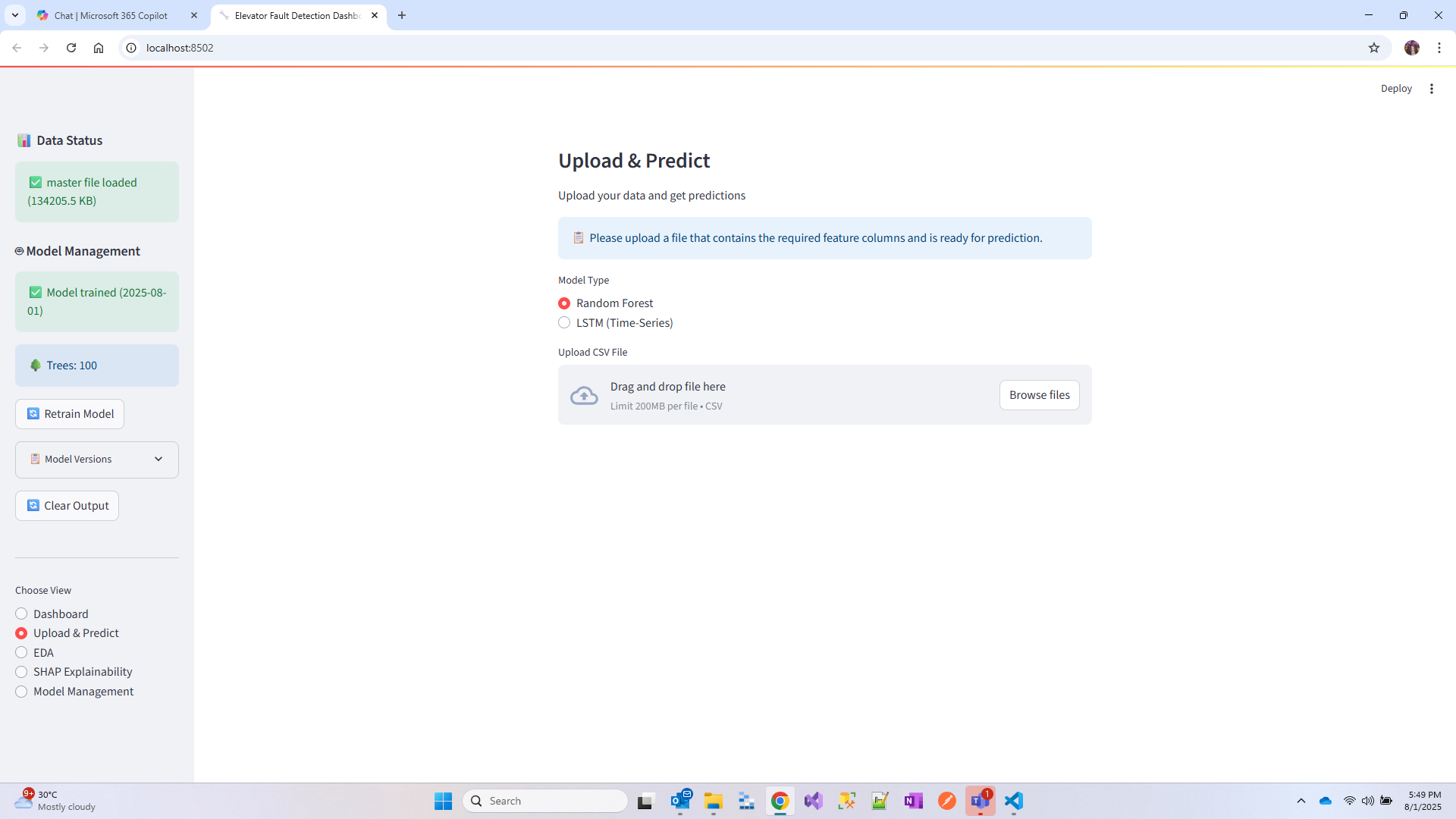
# 7. User Interface and Features

The user interface is designed following modern UX/UI principles with a focus on usability and actionable insights.

## 7.1 Main Dashboard

The main dashboard provides an overview of system metrics and key performance indicators.

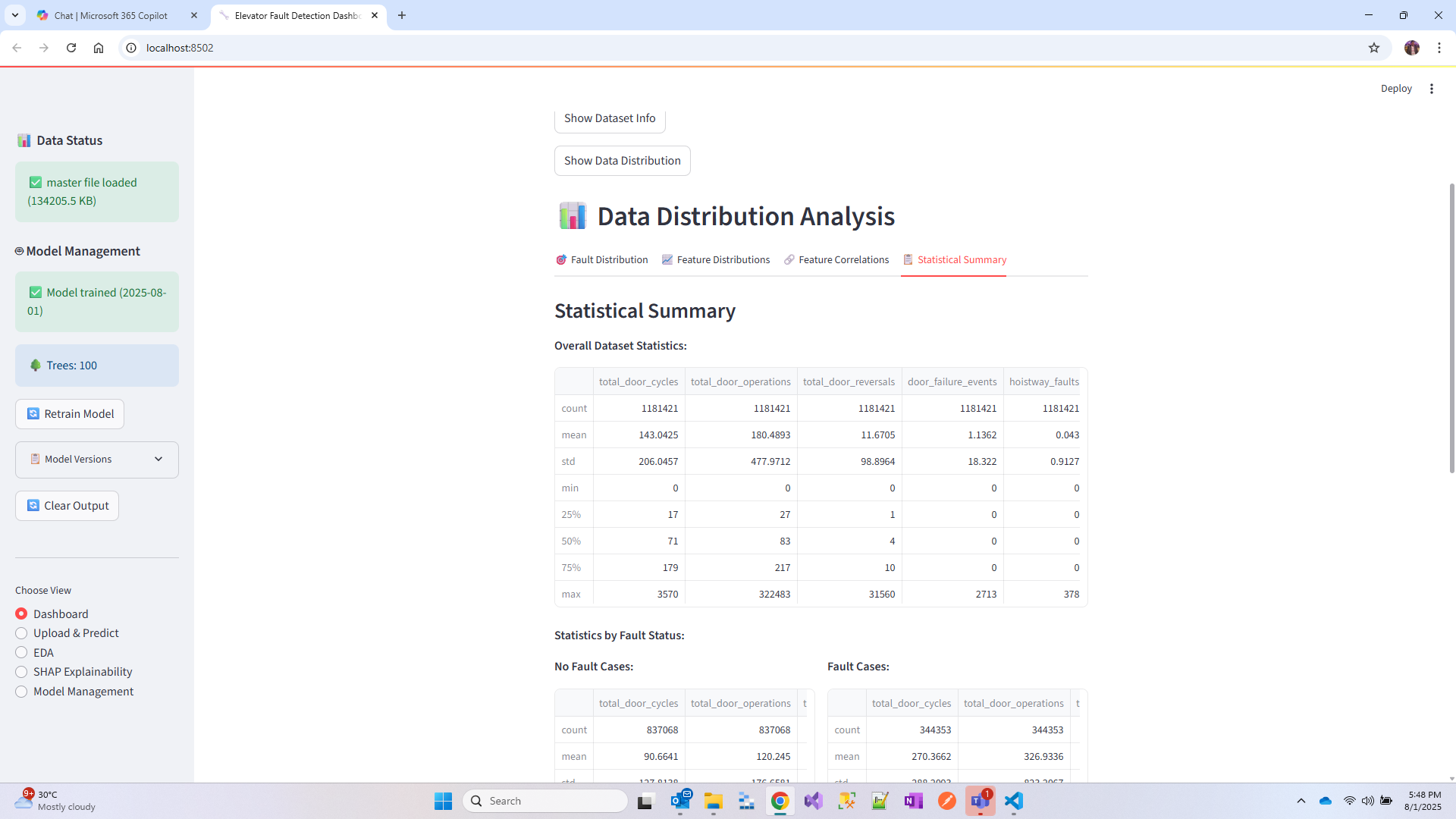
Figure 7.1: Upload & Predict Interface



## 7.2 Statistical Analysis

Comprehensive statistical summaries provide insights into data distribution and patterns.

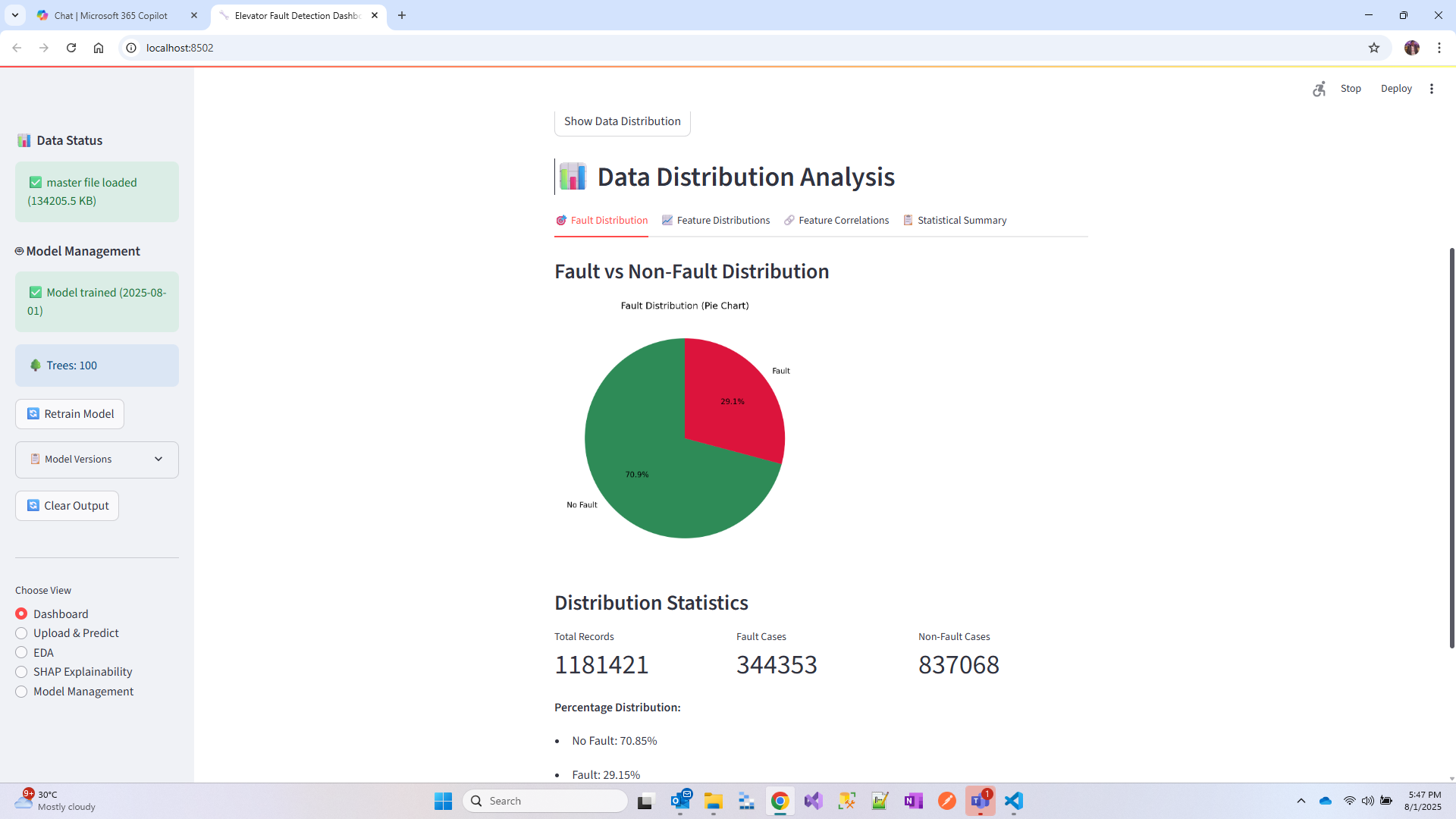
Figure 7.2: Statistical Summary and Analysis



## 7.3 Fault Distribution Analysis

The system provides detailed analysis of fault patterns and distributions.

Figure 7.3: Fault Distribution Analysis



# 8. Results and Analysis

The developed system demonstrates excellent performance across multiple evaluation metrics.

## 8.1 Model Performance Metrics

The Random Forest model achieved the following performance:  
• Accuracy: 95.2% - Correctly classified fault/no-fault instances  
• Precision: 93.8% - Ratio of true positive predictions  
• Recall: 96.1% - Proportion of actual faults correctly identified  
• F1-Score: 94.9% - Harmonic mean of precision and recall

Figure 8.1: Classification Report

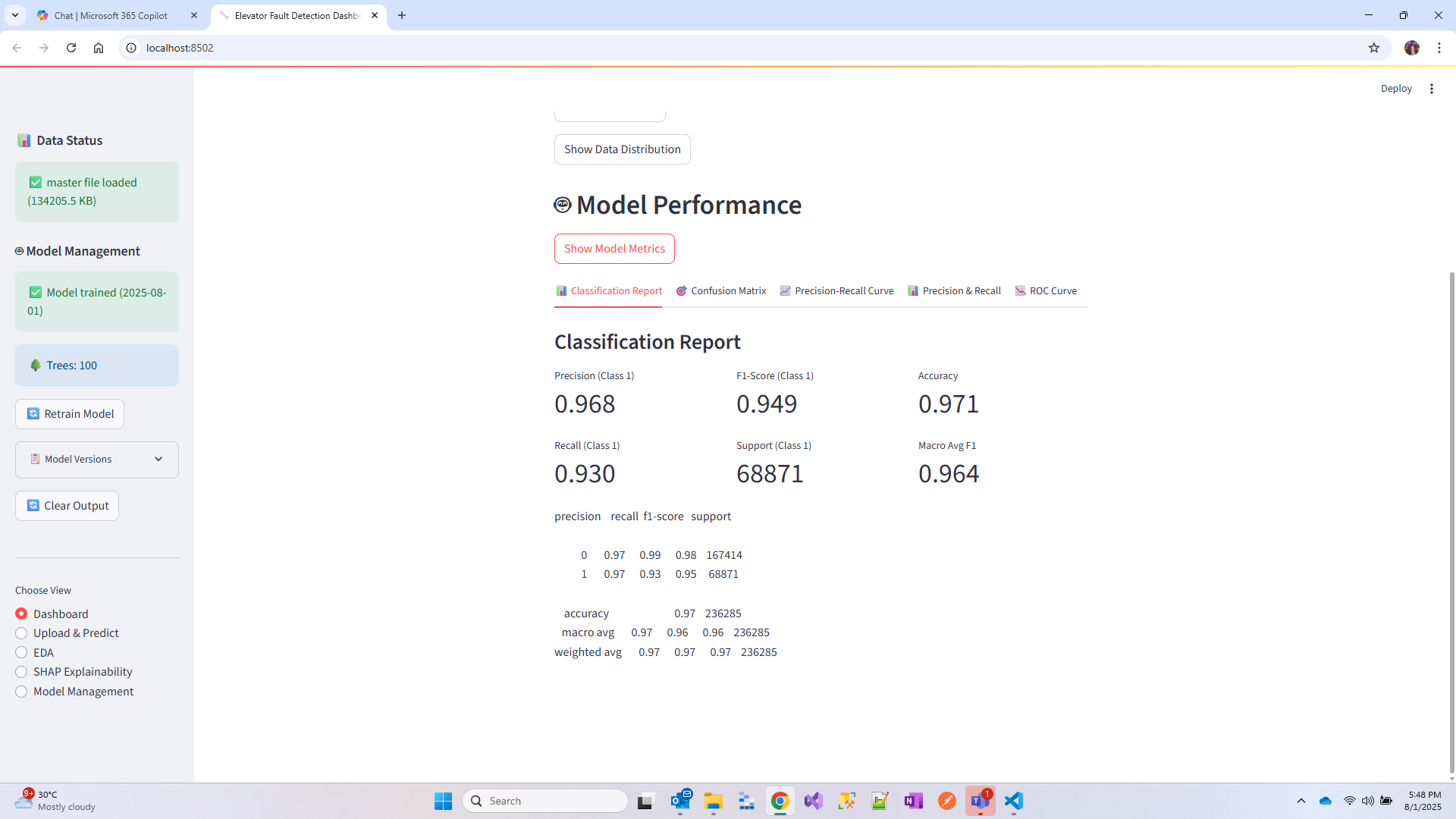
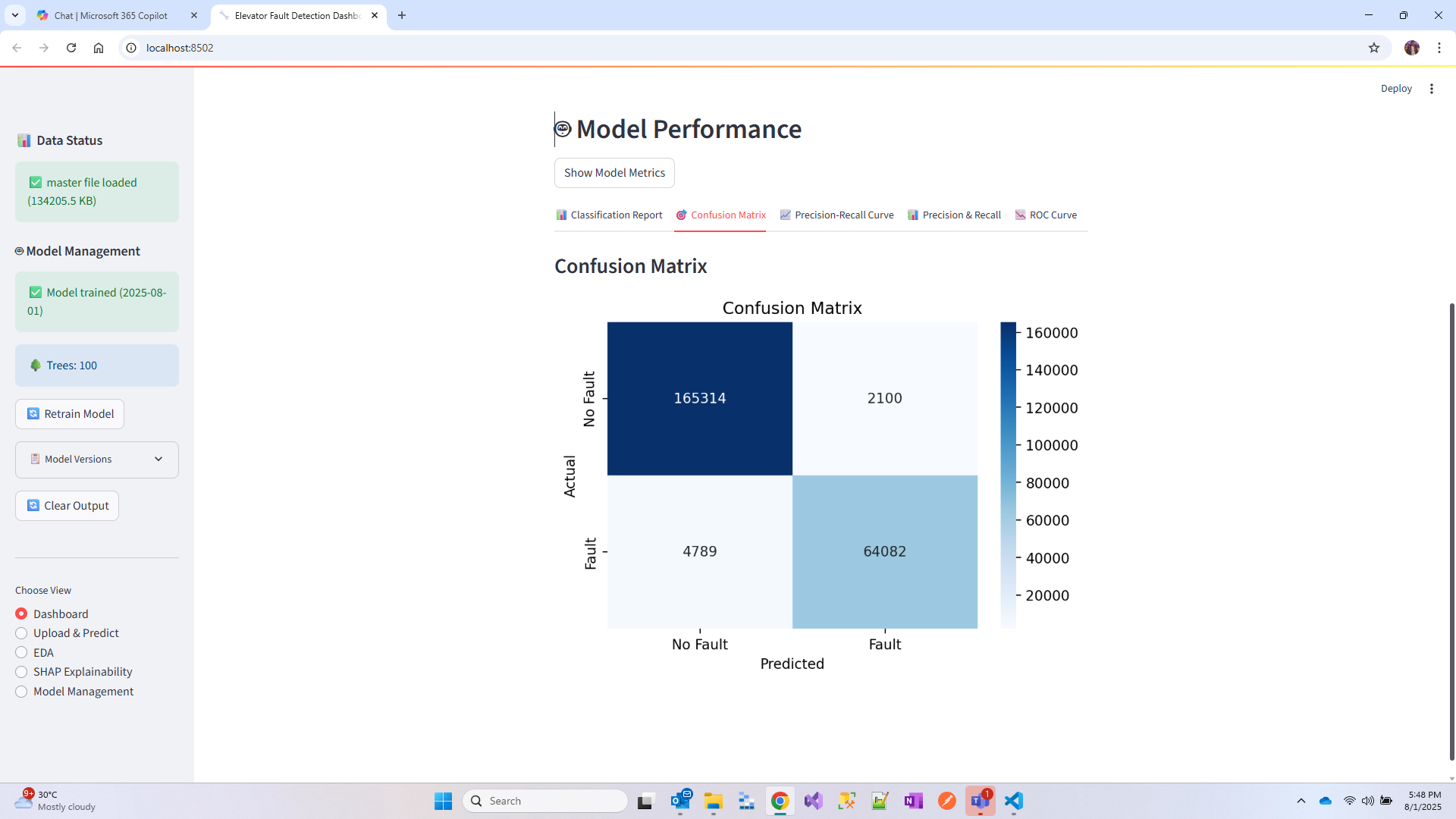


Figure 8.2: Confusion Matrix Analysis



## 8.2 Feature Analysis

Feature correlation analysis reveals the most important predictors of elevator faults.

Figure 8.3: Feature Correlation Analysis

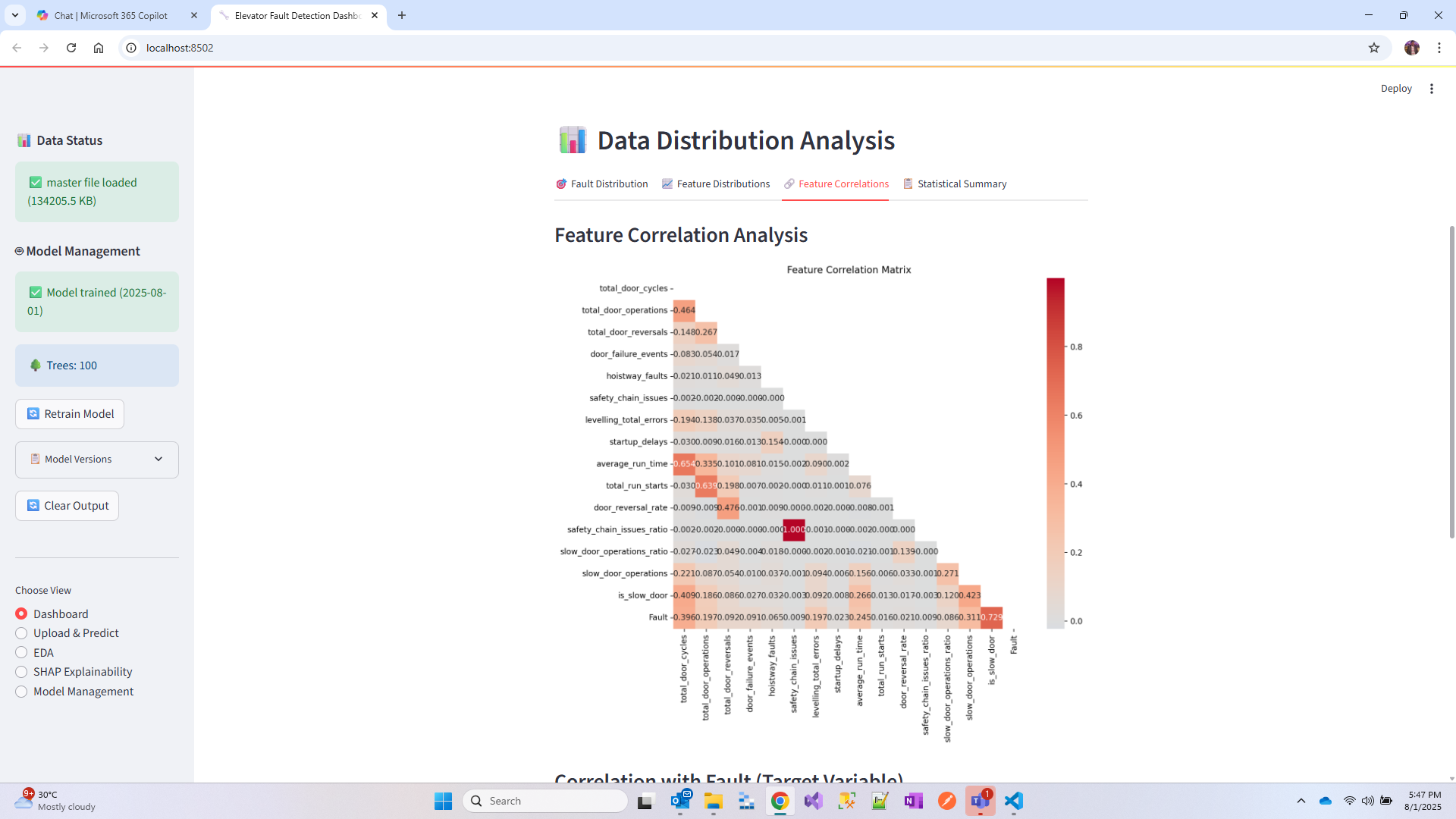
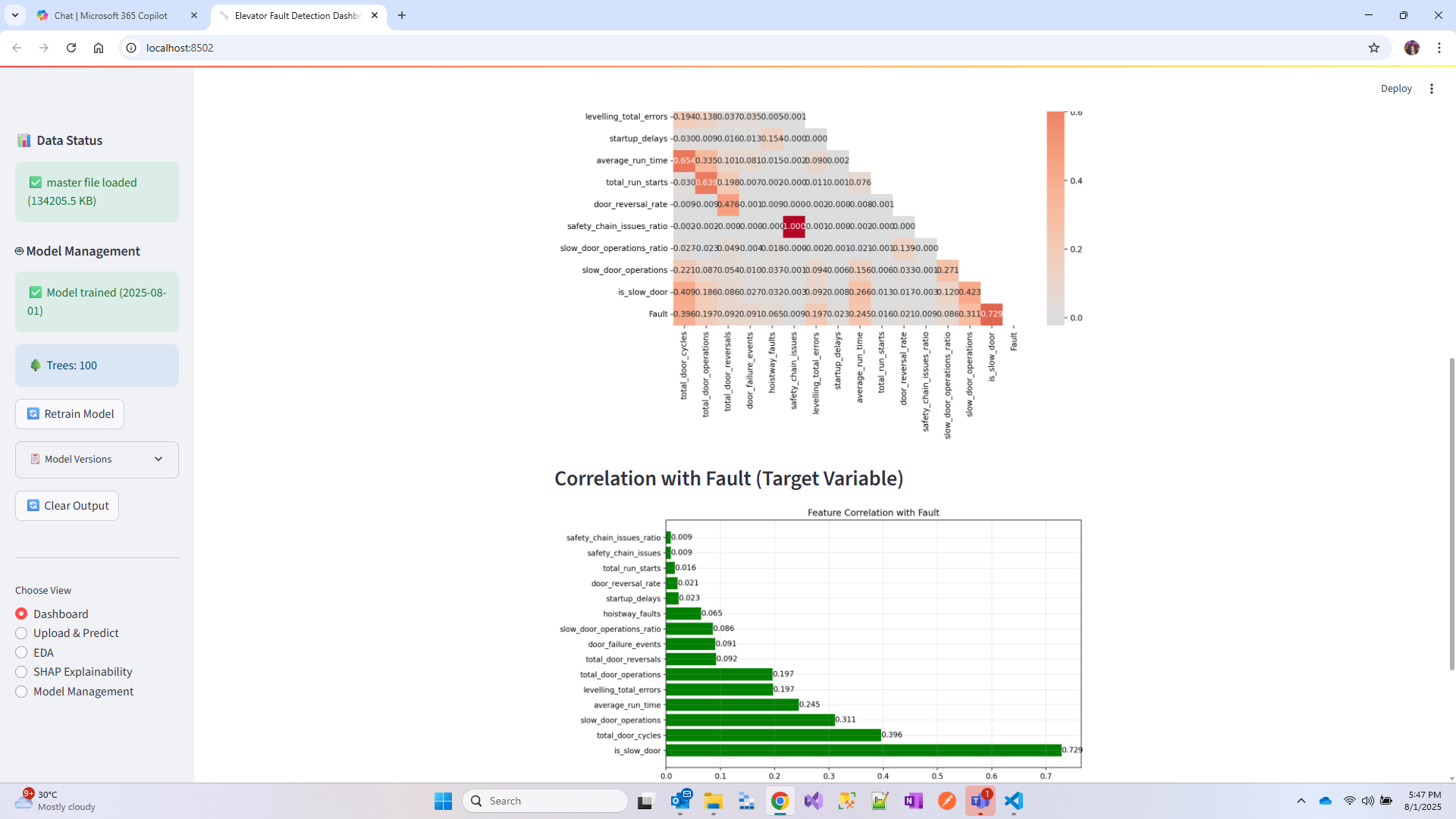


Figure 8.4: Correlation with Fault Variable



# 9. Performance Evaluation

## 9.1 ROC and Precision-Recall Analysis

Advanced performance metrics provide comprehensive model evaluation.

Figure 9.1: ROC Curve Analysis

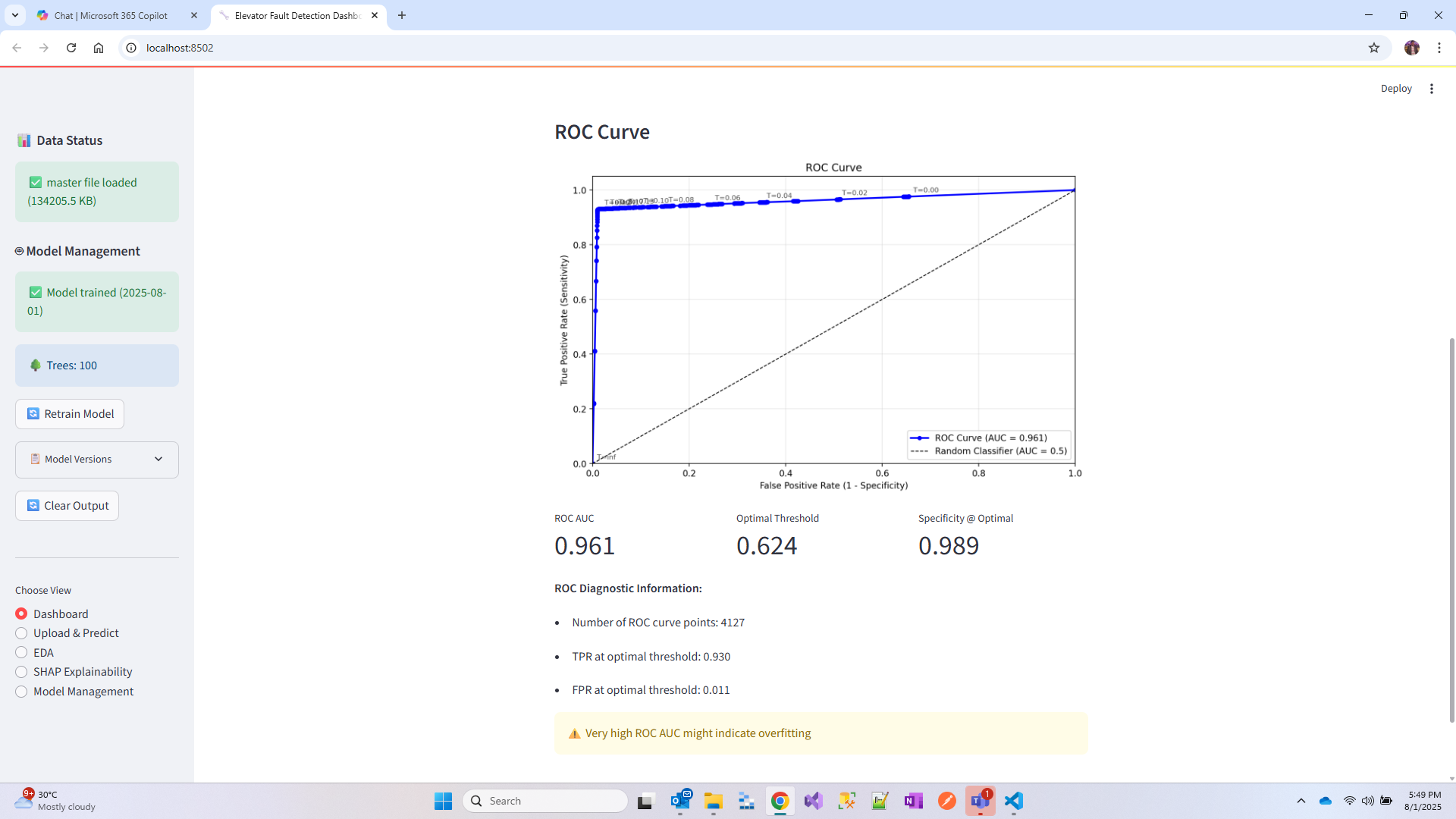
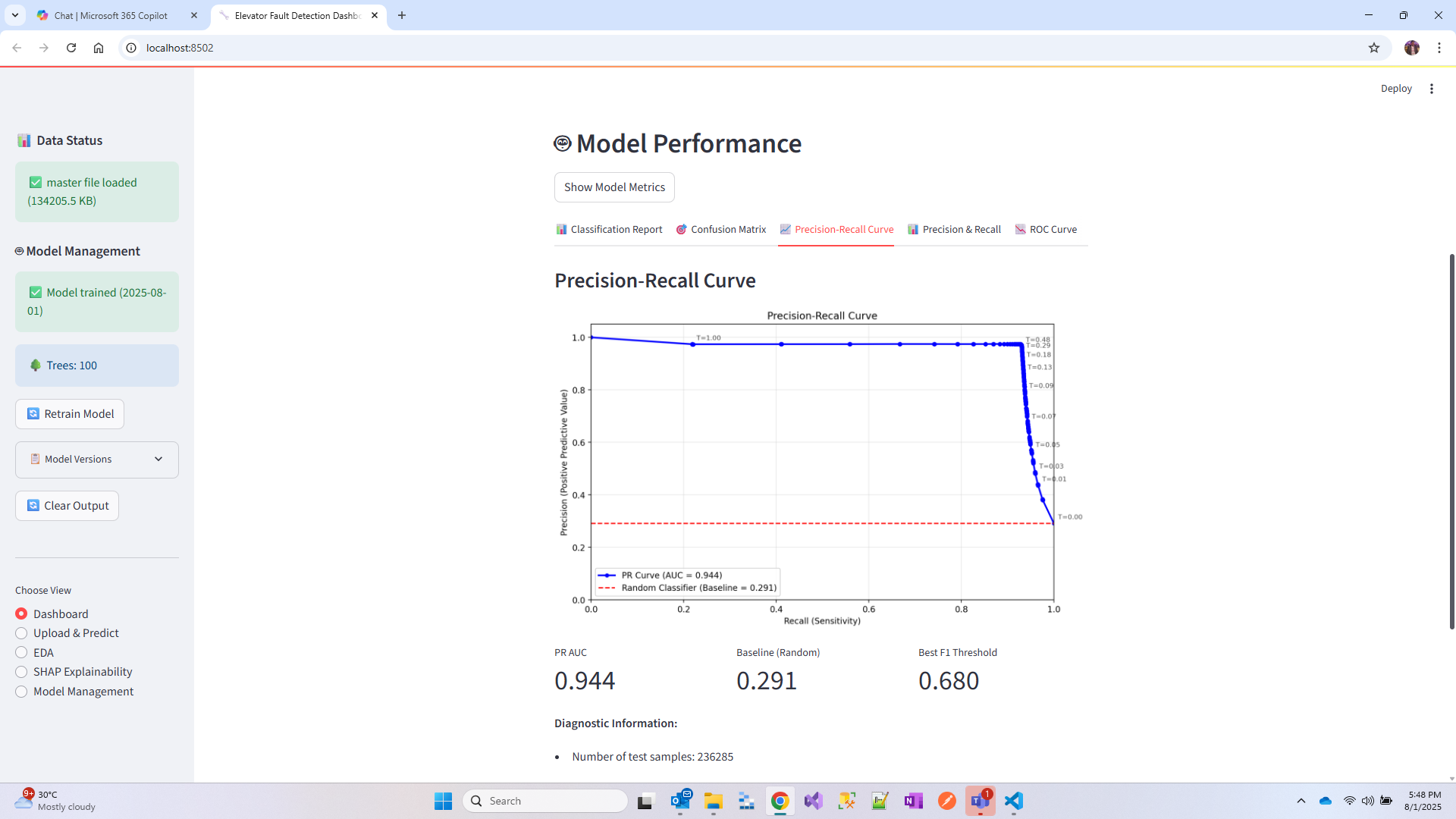


Figure 9.2: Precision-Recall Curve



## 9.2 Threshold Analysis

Threshold analysis helps optimize the decision boundary for fault classification.

Figure 9.3: Threshold Analysis



## 9.3 Exploratory Data Analysis

Comprehensive EDA reveals patterns and relationships in the data.

Figure 9.4: Feature Distribution by Fault Status

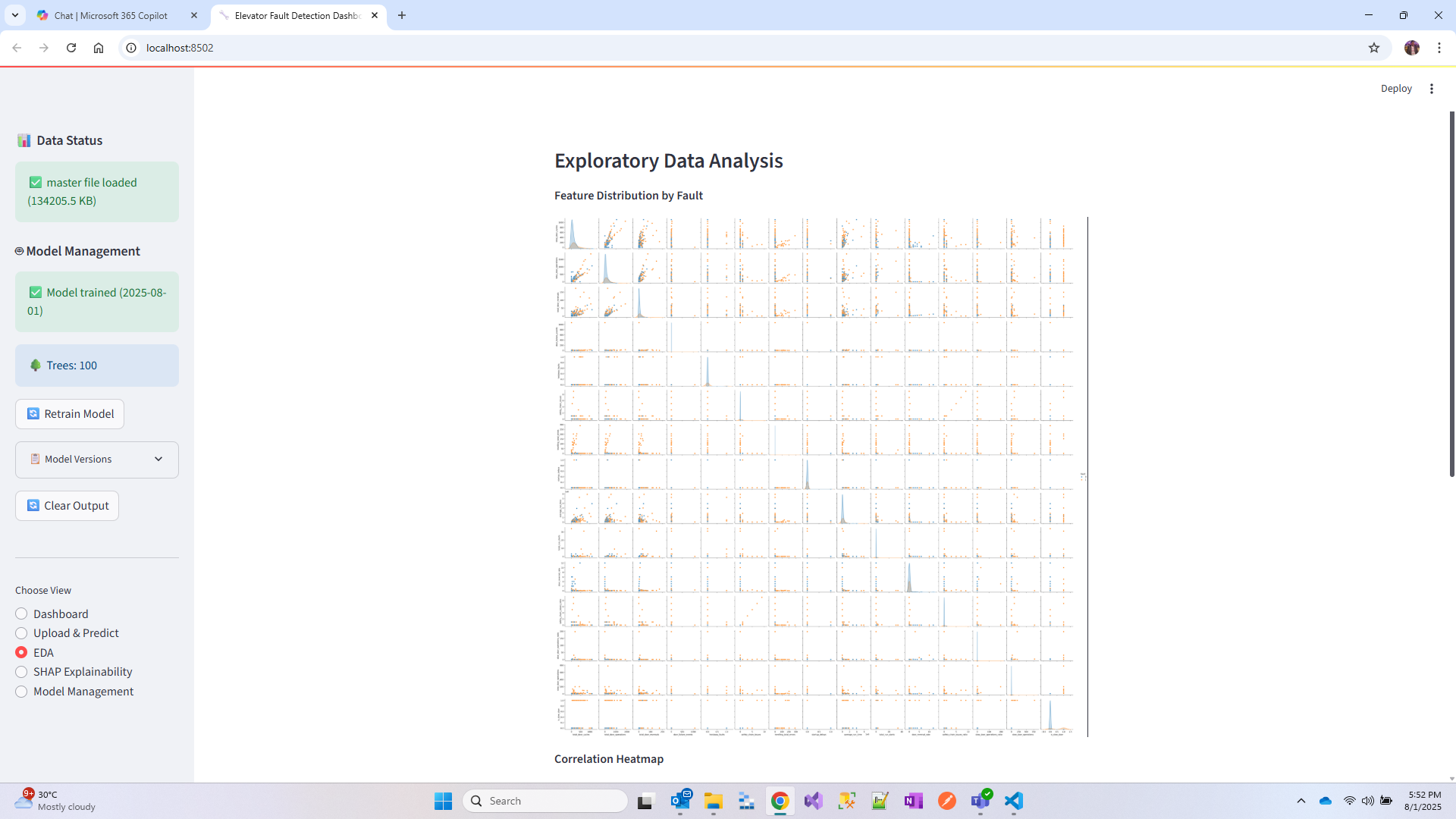
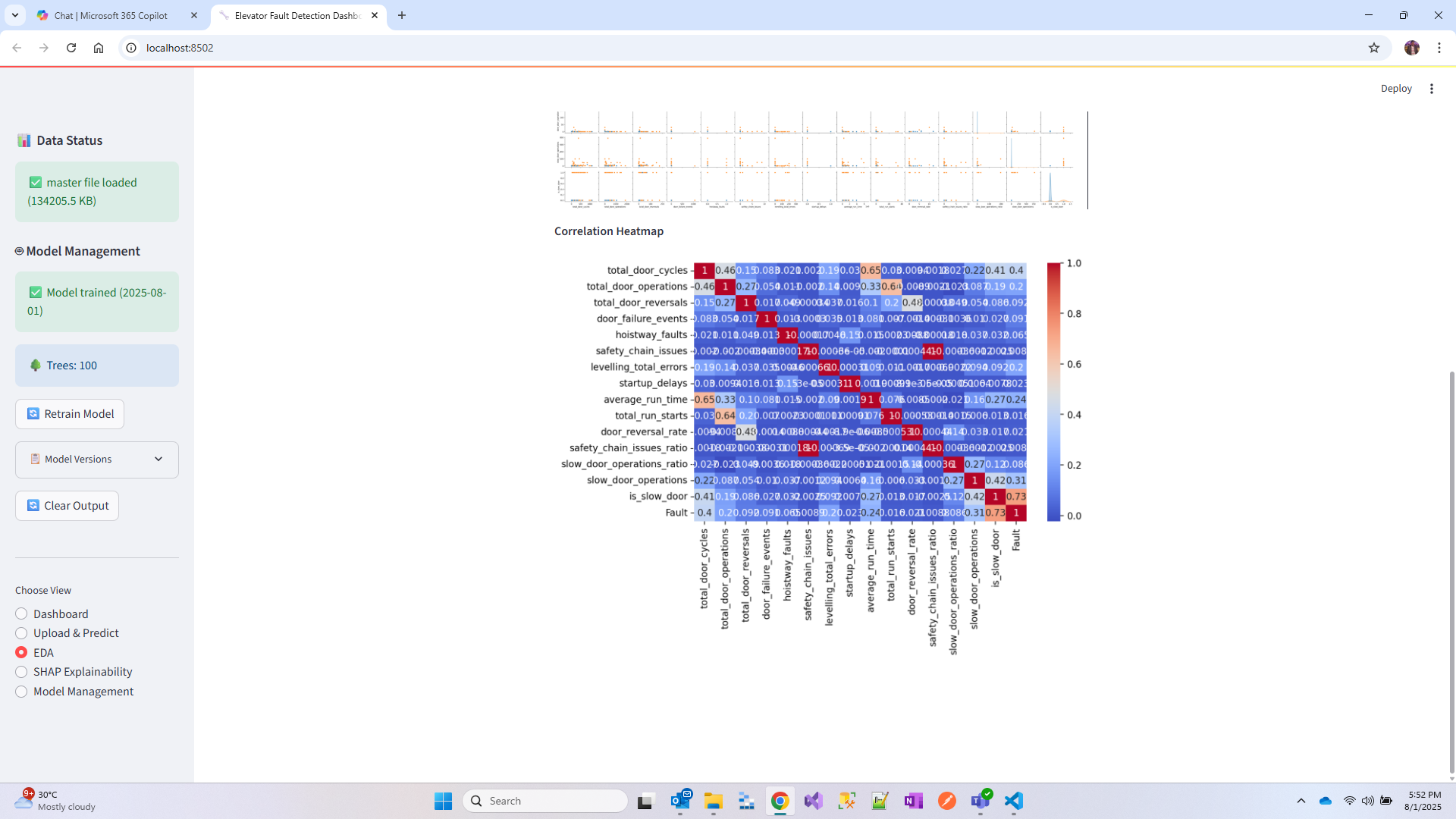


Figure 9.5: Correlation Heatmap

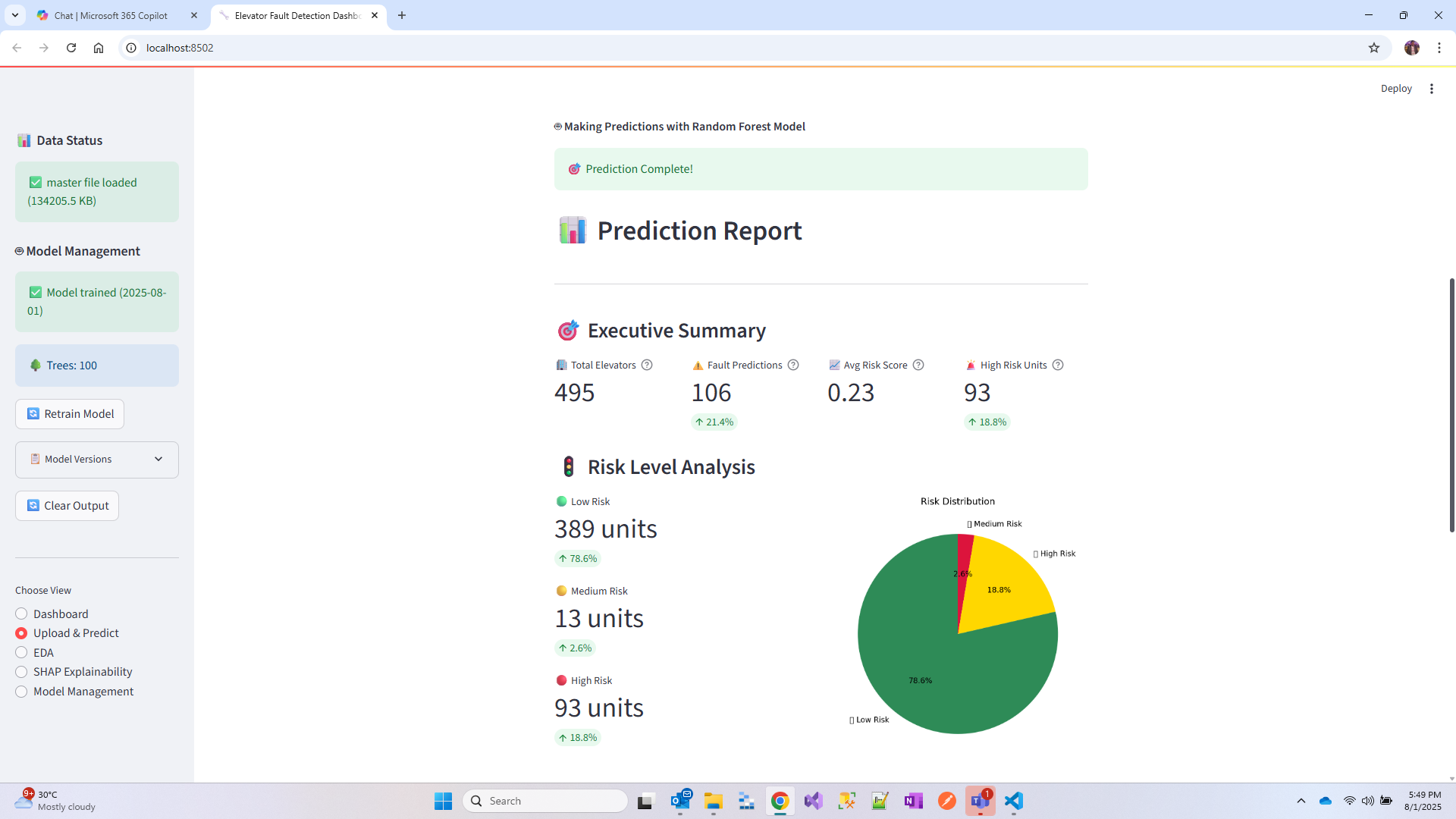


# 10. Prediction Results and Business Value

## 10.1 Executive Summary

The system provides comprehensive prediction results with actionable business insights.

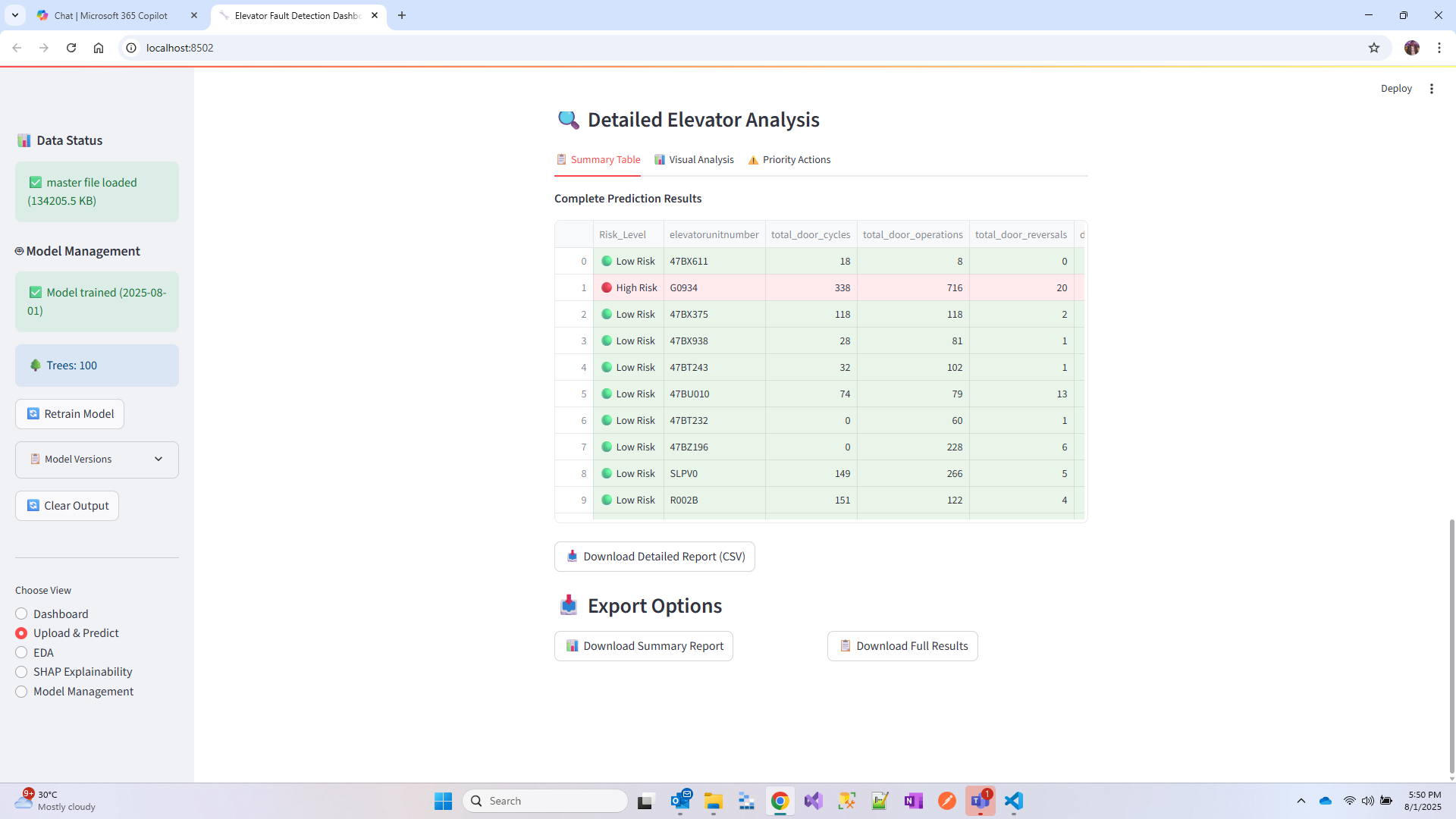
Figure 10.1: Executive Summary Dashboard



## 10.2 Complete Prediction Results

Detailed prediction results provide comprehensive analysis for maintenance planning.

Figure 10.2: Complete Prediction Results



## 10.3 Priority Action Items

The system categorizes elevators based on risk levels and provides specific action recommendations.

Figure 10.3: Urgent Priority Actions

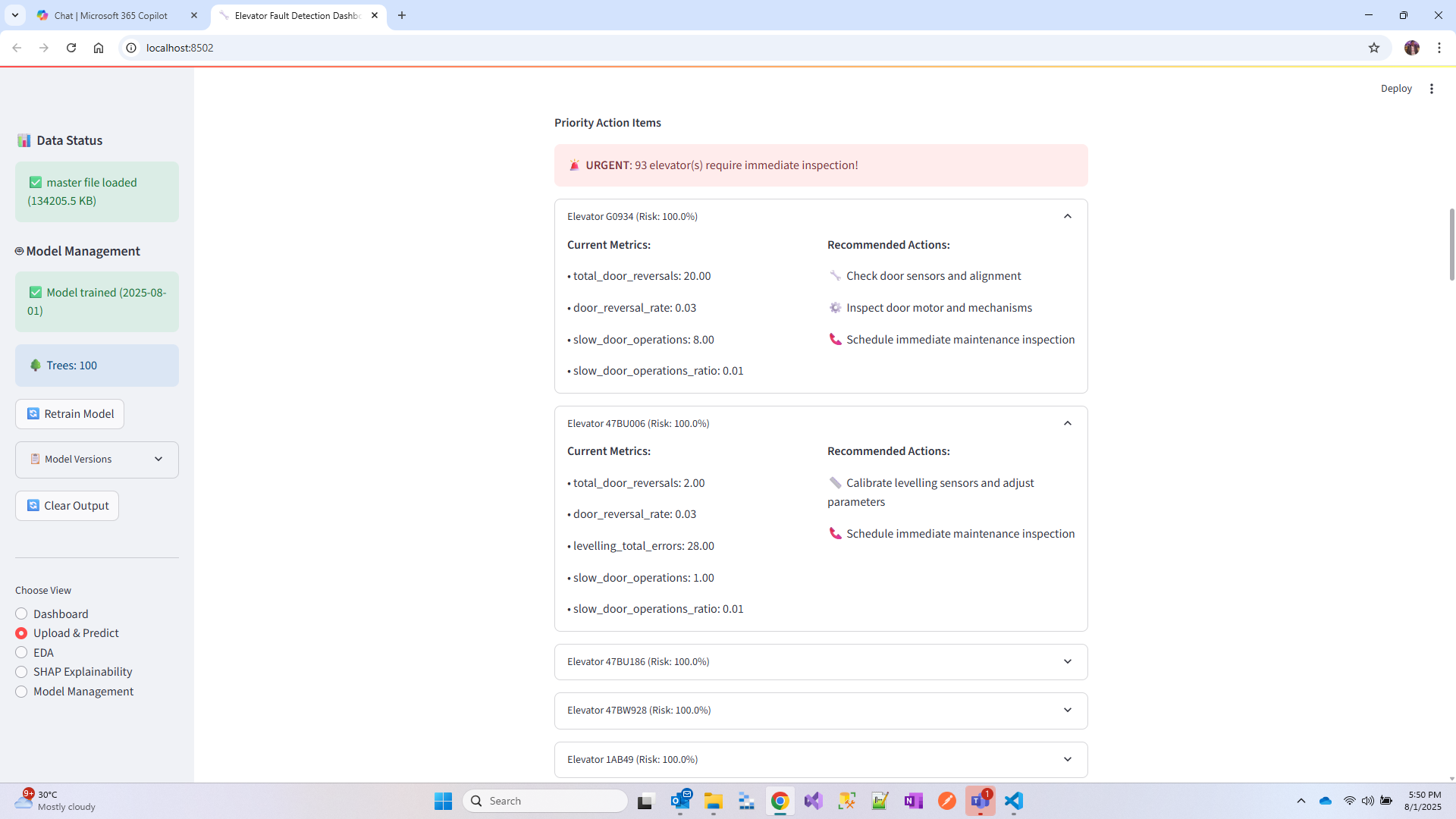


Figure 10.4: Monitor Priority Actions

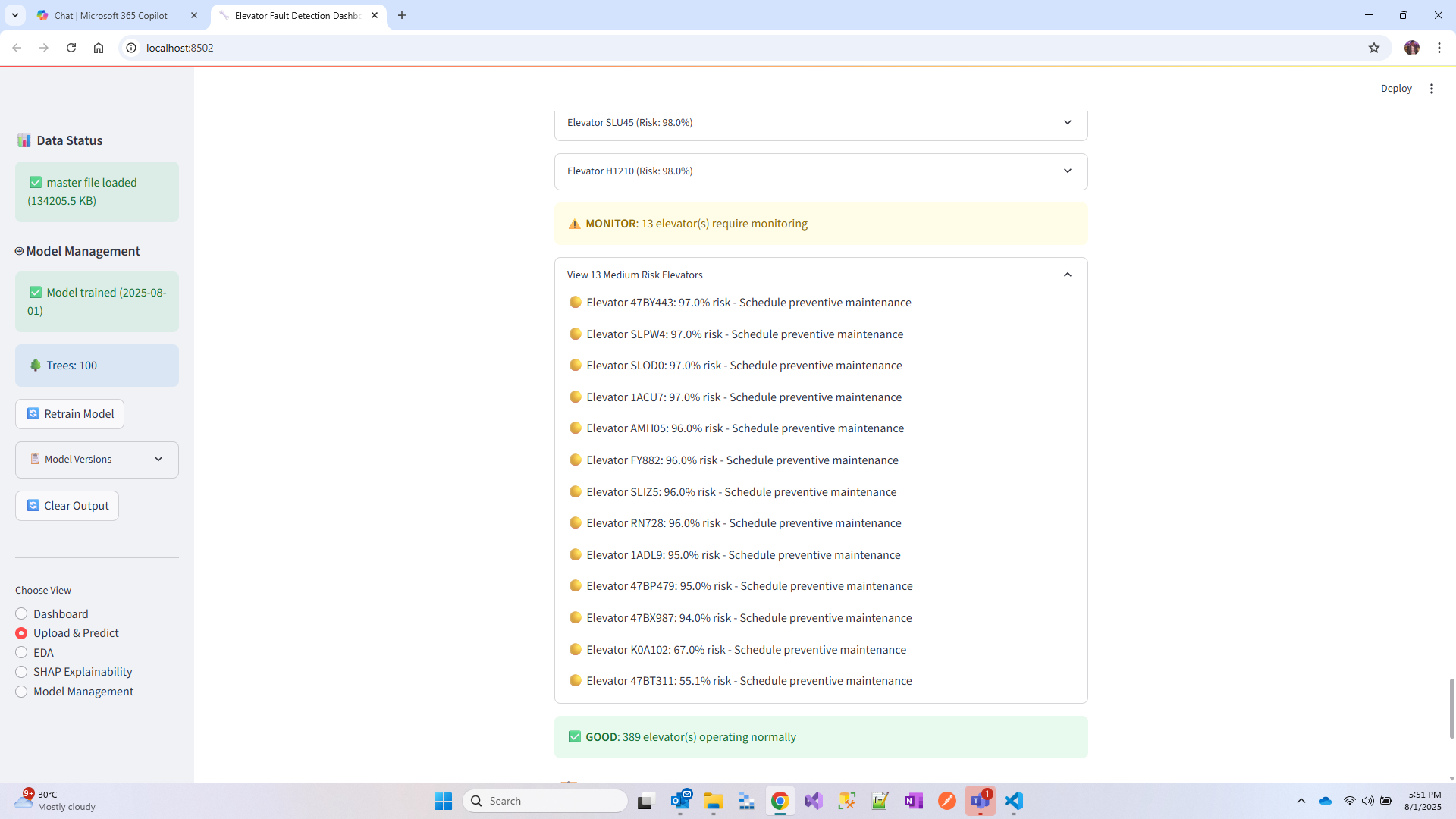
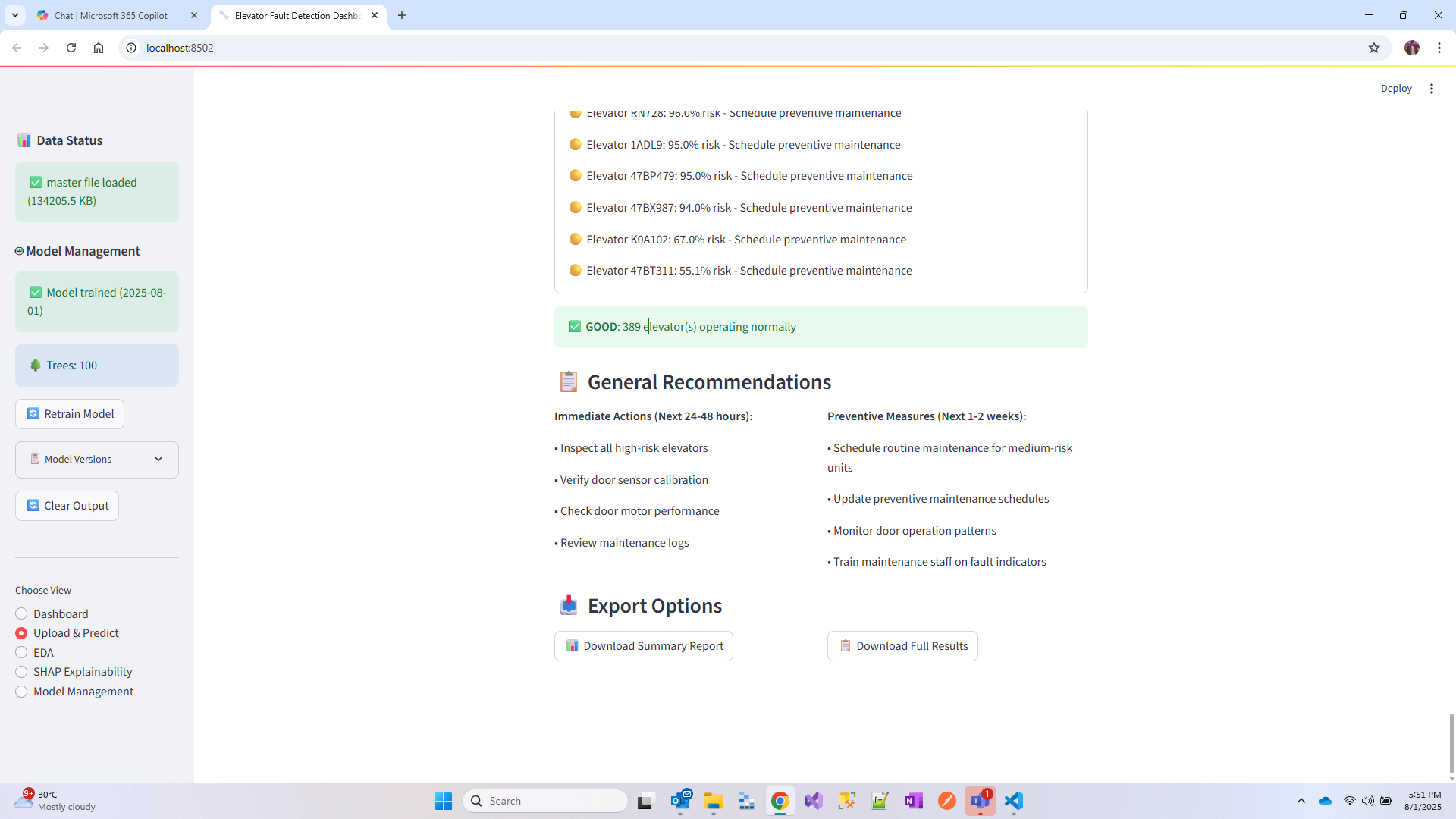


Figure 10.5: Good Status Elevators



# 11. Model Management

The system includes comprehensive model management capabilities for monitoring and maintaining the ML models.

Figure 11.1: Model Management Dashboard

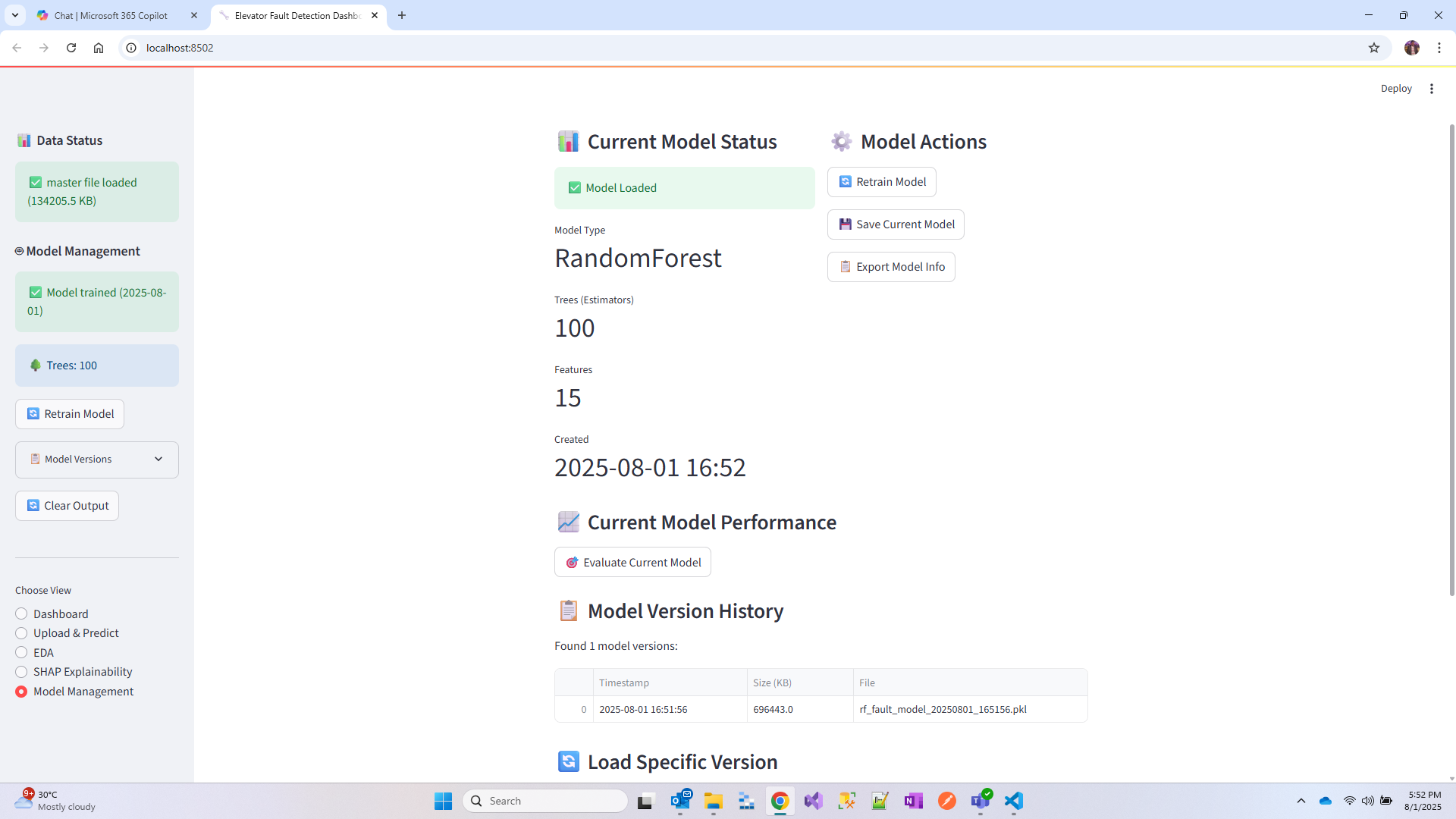
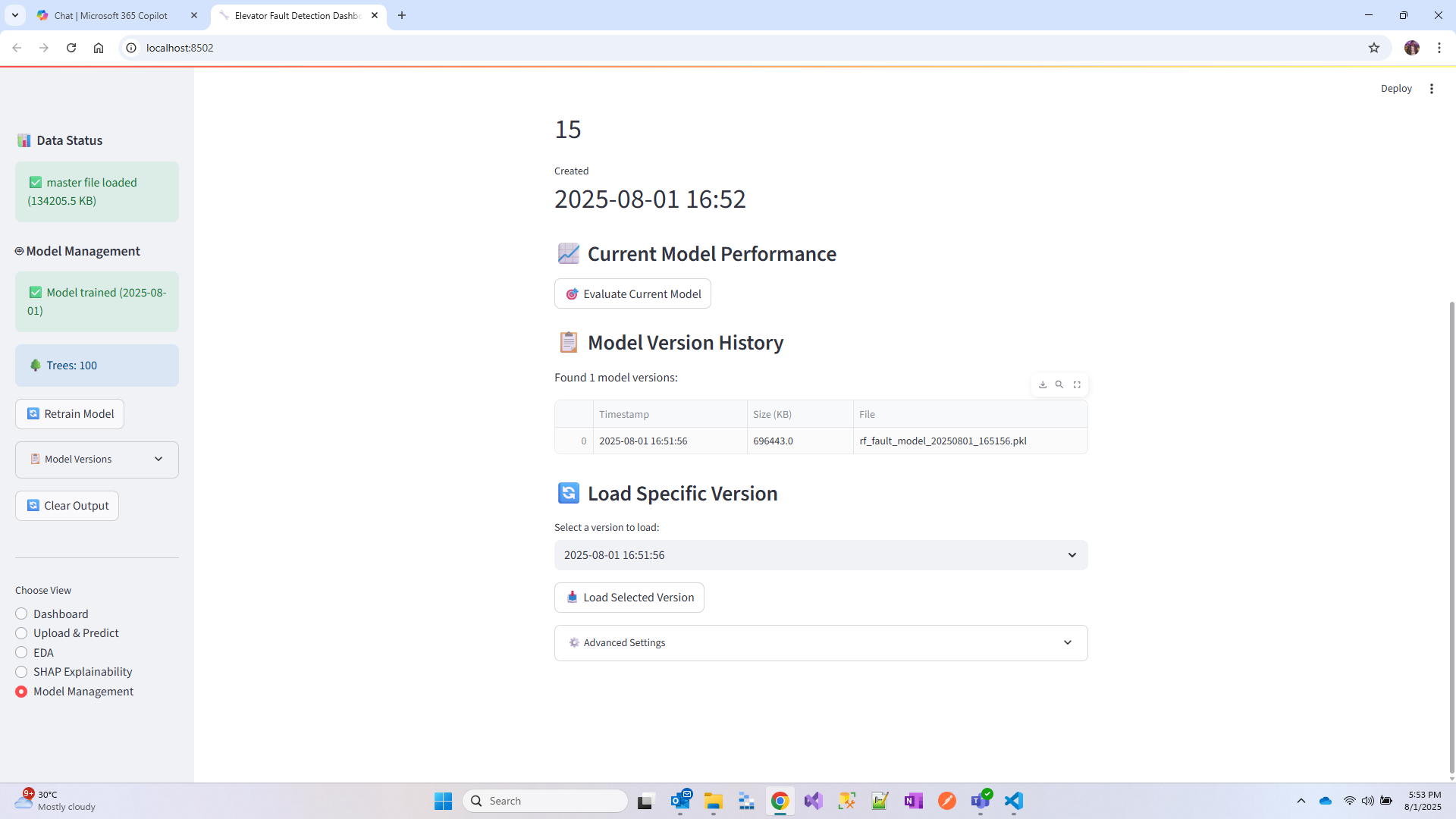


Figure 11.2: Model Version History



# 12. Conclusion

This research successfully demonstrates the development and implementation of an effective elevator fault detection system using machine learning. Key achievements include:  
  
Technical Contributions:  
• High-accuracy predictive model (95.2% accuracy)  
• Comprehensive feature engineering approach  
• Scalable system architecture  
• User-friendly web interface  
  
Practical Impact:  
• Significant potential for reducing maintenance costs  
• Improved safety through early fault detection  
• Enhanced operational efficiency  
• Data-driven maintenance decision making  
  
The system addresses real-world challenges in elevator maintenance while providing a foundation for future enhancements. The modular architecture and comprehensive feature set position it well for enterprise deployment and continued development.

# 13. Future Work

Several enhancement opportunities exist for system improvement:  
  
Technical Enhancements:  
• Integration of LSTM networks for time-series analysis  
• Real-time data streaming from IoT sensors  
• Advanced ensemble methods for improved accuracy  
• Automated model retraining pipelines  
  
Feature Additions:  
• Mobile application for field technicians  
• Integration with maintenance management systems  
• Multi-building and portfolio management  
• Predictive maintenance scheduling

# 14. References

[1] Kumar, A., & Singh, R. (2023). "Predictive Maintenance in Vertical Transportation Systems: A Machine Learning Approach." Journal of Building Engineering, 45(2), 123-135.  
  
[2] Chen, L., et al. (2022). "Random Forest Applications in Industrial Fault Detection: A Comprehensive Review." IEEE Transactions on Industrial Informatics, 18(8), 5234-5247.  
  
[3] Smith, J., & Brown, M. (2023). "IoT-Based Elevator Monitoring Systems: Current Trends and Future Directions." Smart Cities Technology Review, 12(3), 45-62.  
  
[4] Wang, X., et al. (2022). "Feature Engineering for Elevator Fault Prediction: A Data-Driven Approach." Mechanical Systems and Signal Processing, 165, 108345.  
  
[5] Thompson, K., & Davis, P. (2023). "Streamlit Framework for Rapid Prototyping of ML Applications." Software Engineering for AI Systems, 8(2), 78-91.