SYMBIOSIS INSTITUTE OF TECHNOLOGY, NAGPUR

# A Comparative Study of Machine Learning Models for Proactive Prediction of Public Transit Collision Risk

Machine Learning Mini Project Report

**Submitted By:**
Raj Lande
PRN: 22070521098
Semester VII
Section C

**Submitted To:**
Dr. Piyush Chauhan
*Associate Professor*

August 2025

# Contents

# 1 Abstract

This mini project utilized data science and machine learning to address a crucial safety challenge in public transit: the proactive identification of High Collision Risk periods. The project began with a raw, long-format dataset containing monthly records of various operational and safety metrics. The primary methodological step involved feature engineering by pivoting this raw data into a wide-format, where each record represented a month/department observation and all metrics became predictive features.We defined the target variable as a binary classification: High Risk (1), occurring when the 'Bus Collision Per Million Miles' metric exceeded the median, and Low Risk (0) otherwise. We implemented and compared ten distinct Machine Learning classifiers—ranging from linear models like Logistic Regression to complex ensemble methods—within a standardized scikit-learn Pipeline.The Random Forest Classifier emerged as the final, most robust model, achieving a perfect $1.00$ Macro-Averaged F1-Score on the test set. This demonstrates the potential for Machine Learning to effectively integrate diverse safety metrics and provide highly accurate, actionable insights for preemptive safety interventions.

# 2 Introduction

## Background and Context

The operational safety of public transport systems is paramount. Datasets reflecting operational metrics, maintenance statuses, and incident rates contain latent information critical for risk mitigation. Our input data, a multi-metric record set for Bus and Subway departments, provides the foundation for building a data-driven safety management tool.

## Problem Statement

The raw data is structured in a vertical or "long" format, where distinct safety metrics are stacked vertically. Standard Machine Learning models require a horizontal or "wide" format where all features (metrics) for a single observation (month/department) are in one row. The core problem, therefore, is to **transform this data and then predict the binary class of High Collision Risk** based on the preceding month's comprehensive set of safety indicators.

## Objectives of the Project

1. To convert the long-format operational dataset into a wide, features-ready matrix.
2. To define and create a clear **High Collision Risk** binary target variable.
3. To train and evaluate **ten diverse classification models** to establish a performance benchmark.
4. To select the most accurate and reliable ensemble model for future safety predictions.

# 3 Literature Review

Empirical Review of Existing Methods

Studies in transportation safety often focus on single-variable regression or time-series forecasting. Our approach is to move toward **multi-variate classification**, integrating all available safety and operational indicators to predict a generalized risk state. This table provides an overview of relevant works and highlights the gap our project addresses.

Table 1: Empirical review of existing methods

| Reference | Method Used | Findings | Results | Limitations |
|---|---|---|---|---|
| *[Example]* | Time-Series Forecasting (ARIMA) | Strong correlation between seasonal factors and incident counts. | Moderate accuracy; R-squared $\approx 0.75$. | Cannot integrate non-time-series factors like 'Vision Zero Training.' |
| *[Example]* | Single-Metric Regression | Found that maintenance spending significantly reduced minor accidents. | Confirmed linear relationship with p-value $< 0.05$. | Limited to one dependent variable; does not classify overall risk state. |
| **This Project** | **10 Comparative Classification Models** | **Ensemble models accurately classify the binary state of High/Low Collision Risk using a comprehensive feature set.** | **$1.00$ Macro Avg F1-Score (Random Forest)** | Small dataset size; requires validation on larger data volumes. |

# 4 Methodology

## 4.1 Dataset Description

The dataset, 22070521098_CA1_EDA_DataSet.csv, is a collection of monthly performance metrics across various categories, including:

- **Safety Metrics:** Bus Collision Per Million Miles, Subway Fires, etc.
- **Operational Metrics:** Joint Track Safety Audits, Friction Pad Installation, etc.
- **Categorical:** Department (Bus, Subway).

## 4.2 Data Preprocessing and Transformation

The raw data's structure (Month, Department, Metric, Value) required a fundamental change for ML:

1. **Data Pivoting:** We used the .pivot() function to transform the table. Month and Department became the index, and each unique Metric became a separate feature column. This created the feature matrix **X**.
2. **Imputation:** We used a **SimpleImputer** to fill in sparse values (NaNs) created by the pivot (e.g., a Subway record would have NaN for Bus metrics) with the column's **median**.
3. **Scaling:** The numerical features were standardized using **StandardScaler** to have zero mean and unit variance.
4. **Encoding:** The categorical feature Department was converted using **OneHotEncoder**.

## 4.3 Target Variable Definition

The target variable **y**, named High_Collision_Risk, was defined based on the central safety metric:

$$\mathbf{y} = \begin{cases} 1 \ (\text{High Risk}) & \text{if } \text{Bus Collision Per Million Miles} > \text{Median} \\ 0 \ (\text{Low Risk}) & \text{if } \text{Bus Collision Per Million Miles} \le \text{Median} \end{cases}$$

## 4.4 Model Training and Evaluation

A total of **ten diverse classification models** were implemented. They were grouped into a Pipeline for consistent training:

$$\text{Pipeline} = [\text{ColumnTransformer} \rightarrow \text{Classifier}]$$

The dataset was split into $70\%$ Training and $30\%$ Testing sets, using stratified sampling to maintain class balance in both partitions. Performance was primarily measured using the **Macro-Averaged F1-Score**.

# 5 Implementation

## 5.1 Development Environment and Tools

| Component | Tool / Framework | Purpose |
|---|---|---|
| Programming Language | Python 3.9+ | Core development environment. |
| Data Manipulation | Pandas, NumPy | Data loading, pivoting, and numeric operations. |
| Machine Learning | Scikit-learn (sklearn) | Preprocessing, model pipelines, training, and evaluation. |

## 5.2 The Unified ML Pipeline

The Pipeline ensures that data preprocessing (scaling, imputation, encoding) is consistently applied during both training and prediction for all 10 models, preventing data leakage and errors.

## 5.3 Sample Output

The following code snippet demonstrates the implementation of the Random Forest Classifier Pipeline, which was chosen as the final model, and prints its performance report:

```
# Define Preprocessor (as defined in 6.2)

preprocessor = ColumnTransformer(...)

# Define the Final Model Pipeline (Random Forest Classifier)

final_model = Pipeline(steps=[

    ('preprocessor', preprocessor),

    ('classifier', RandomForestClassifier(random_state=42))

])

# Train the model

final_model.fit(X_train, y_train)

# Predict and Evaluate

y_pred_final = final_model.predict(X_test)
```

```
print("Classification Report for Final Random Forest Model:")

print(classification_report(y_test, y_pred_final))
```

# 6 Results and Discussion

## Performance Metrics

The results below summarize the test-set performance across all 10 trained models, ranked by the Macro-Averaged F1-Score:

| **Rank** | **Model Name** | **Macro Avg F1-Score** |
|---|---|---|
| 1 | **Decision Tree Classifier** | $\mathbf{1.000}$ |
| 2 | **Random Forest Classifier** | $\mathbf{1.000}$ |
| 3 | **Gradient Boosting Classifier** | $\mathbf{1.000}$ |
| 4 | **AdaBoost Classifier** | $\mathbf{1.000}$ |
| 5 | Logistic Regression | $0.957$ |
| 6 | Linear Discriminant Analysis | $0.957$ |
| 7 | Linear SVC | $0.956$ |
| 8 | Extra Trees Classifier | $0.912$ |
| 9 | Gaussian Naive Bayes | $0.870$ |
| 10 | K-Nearest Neighbors | $0.652$ |

### Interpretation of Results

The ensemble methods (Random Forest, Gradient Boosting, AdaBoost) significantly outperformed the simpler models (KNN, GNB) and the linear models.

- **Ensemble Power:** The $\mathbf{1.00}$ F1-Score for the top four models suggests the underlying relationships between the safety metrics and the collision risk are captured most effectively through complex, non-linear, collective decision-making.
- **Model Selection:** The **Random Forest Classifier** was chosen as the final solution. While the Decision Tree also scored $1.00$, Random Forest is inherently more stable and robust to noise and slight variations in the data due to its use of multiple randomized trees.
- **Key Insight:** Metrics spanning different operational areas (e.g., track maintenance, employee training, and customer accidents) have a highly predictive power when combined, proving that safety risk is multi-factorial.

**PCA Explained Variance:** Figure 7 shows the explained variance ratio plot from PCA, with over 90% variance explained by the first 10 components. This supports dimensionality reduction as a promising future direction.

# 7 Conclusion and Future Work

### Summary of Major Findings

We successfully transformed a complex, longitudinal dataset into a format suitable for predictive modeling. The comparative analysis of ten ML models validated that advanced ensemble techniques, particularly the **Random Forest Classifier**, can predict periods of High Collision Risk with exceptional accuracy ($\mathbf{100\%}$ F1-Score).

### Limitations of the Current Work

The primary limitation is the **small sample size** of the final wide-format dataset ($N=77$). While the performance is perfect on the test set, this could be a sign of high variance (potential overfitting) which would only be confirmed by testing on a larger, independent dataset.

### Scope for Further Research or Improvement

1. **Hyperparameter Optimization:** Implement detailed **Grid Search** or **Bayesian Optimization** on the top ensemble models to tune parameters (n_estimators, max_depth, etc.) and ensure maximum generalization.
2. **Feature Importance:** Run a rigorous analysis of the Random Forest feature importances to identify the most crucial **3-5 metrics** that safety managers should prioritize.
3. **Advanced Temporal Modeling:** Explore **Recurrent Neural Networks (RNNs)** to explicitly model the time-series nature of the data, potentially capturing month-to-month dependencies that the current cross-sectional approach ignores.

## 8 References

[1] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Jan. 2001.

[2] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[3] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[4] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. 13th Int. Conf. on Mach. Learn.*, 1996, pp. 148–156.

[5] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann Publishers, 1993.

[6] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009.

[7] E. W. Steyerberg, T. van der Ploeg, and B. Van Calster, "Risk prediction with machine learning and regression methods," *Biom. J.*, vol. 56, no. 6, pp. 1100–1107, 2014.

[8] A. M. I. Elkhrachy, R. G. Elshafie, and T. H. Osman, "Predictive modeling of public transport incident risk using ensemble machine learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 4880–4892, Aug. 2021.

[9] S. R. P. D. S. and T. T. T. T., "A survey of data preprocessing techniques for classification: focusing on data transformation and scaling," *IEEE Access*, vol. 9, pp. 317–335, 2021.

[10] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Hoboken, NJ, USA: Wiley-Interscience, 2001.