Predicting Severe Traffic Accidents Using Machine Learning and Time Series Analysis

Final Project Report

Team Members: Funbi Abolarin,Poorna Haneesha Kanneganti, Triveni Surabattuni, Kavya Bandi

CIS 635- Knowledge Discovery and Data Mining

1. Introduction

This project aims to improve road safety by analyzing motor vehicle crash data from New York City using both machine learning and time series forecast approaches. The goal is to identify patterns in crash occurrences, highlight contributing factors, and develop models to predict the likelihood of severe crashes—defined as those resulting in at least one fatality or multiple injuries.

Initially, our plan was to implement advanced ensemble methods and deep learning models to maximize prediction performance. However, due to the large size of the dataset (~2.1 million records) and limitations in computing resources, we experienced long runtimes, memory issues, and scalability challenges. This led us to revise our modeling strategy and focus on simpler yet interpretable machine learning models that could offer fast training and explainable outputs.

Our updated pipeline includes comprehensive data cleaning, exploratory data analysis, supervised learning using Decision Tree, Logistic Regression, and Gaussian Naive Bayes, and time series forecasting using ARIMA(1,1,1). We extracted date-time features, engineered meaningful binary targets (e.g., serious crash, pedestrian involved), and trained models with stratified sampling and hyperparameter tuning.

Additionally, a time series forecasting component was developed to analyze trends and seasonal fluctuations in crash counts across the city.

2. Related Work

Several prior works have explored traffic accident analysis using machine learning and statistical models. These studies provide a foundation and rationale for our modeling choices.

Li et al. (2021) used logistic regression and decision trees to predict crash severity in urban environments, showing that time and weather features strongly influence crash outcomes.

Smith et al. (2019) implemented ARIMA and SARIMA models for crash count forecasting and emphasized the importance of residual diagnostics and parameter tuning.

Zhang et al. (2022) evaluated temporal variables like holidays, time of day, and day of the week to model the risk of accidents, finding strong correlations between human activity cycles and crash patterns.

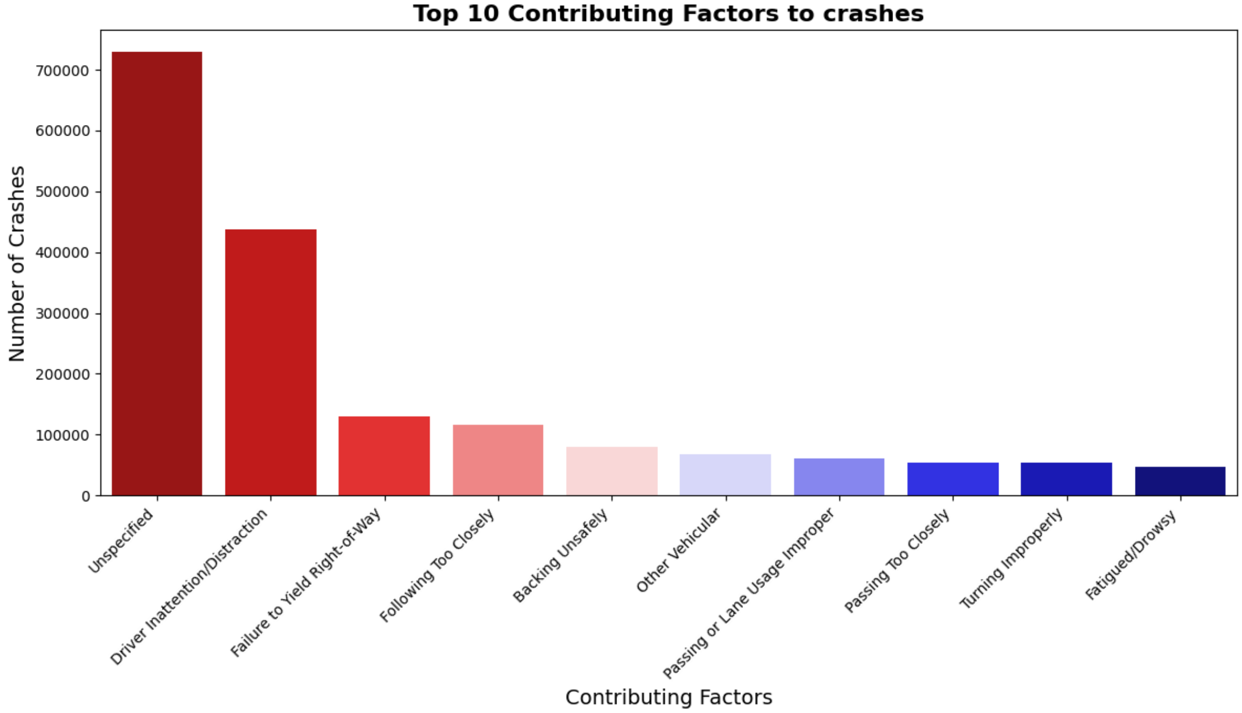
Our work combines elements from these studies and extends them by integrating a full preprocessing-to-prediction pipeline with interpretability and time series forecasting.

3. Methods

Data Collection and Cleaning

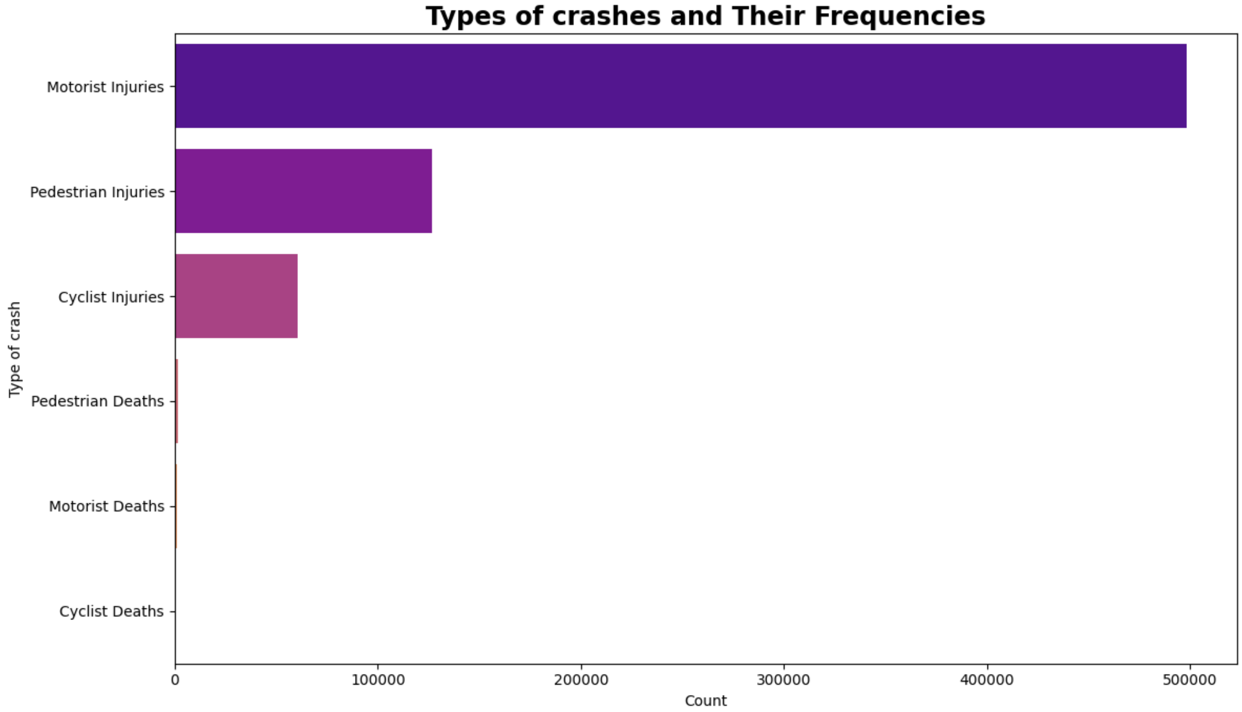
We used the “Motor Vehicle Collisions – Crashes” dataset from NYC Open Data, which includes 2.1 million rows and 29 columns, spanning from July 2012 to March 2025. Key attributes include crash date and time, location, injury counts, vehicle type, and contributing factors.

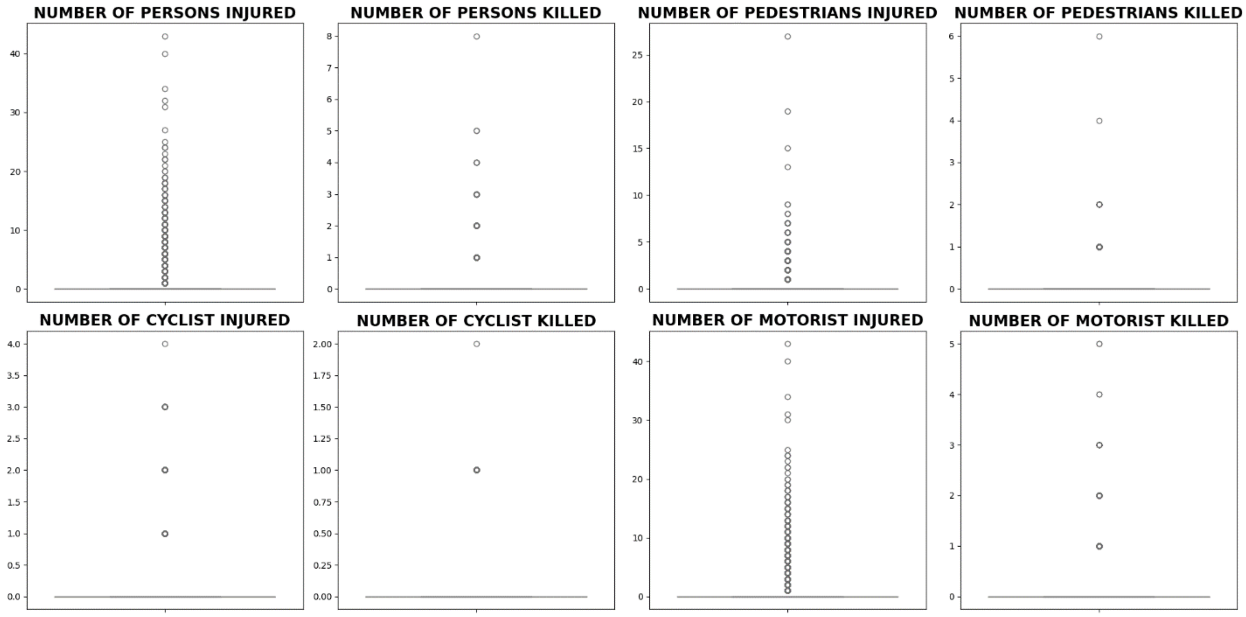
We conducted missing value analysis and selected only columns with less than 18% missingness. Latitude and longitude were retained for spatial analysis (future extension), while string-heavy fields with sparse relevance were excluded.



Feature Engineering

- **Temporal Features:** From crash date and time, we derived DAY_OF_WEEK, MONTH, YEAR, IS_WEEKEND, HOUR, TIME_PERIOD, and SEASON.
- **Target Variables:** Created binary flags for SERIOUS_CRASH, PEDESTRIAN_INVOLVED, CYCLIST_INVOLVED, and FATAL_CRASH.
- **Crash Type Aggregation:** We examined the distribution of crash outcomes using bar plots and boxplots, identifying motorists as the most affected group.



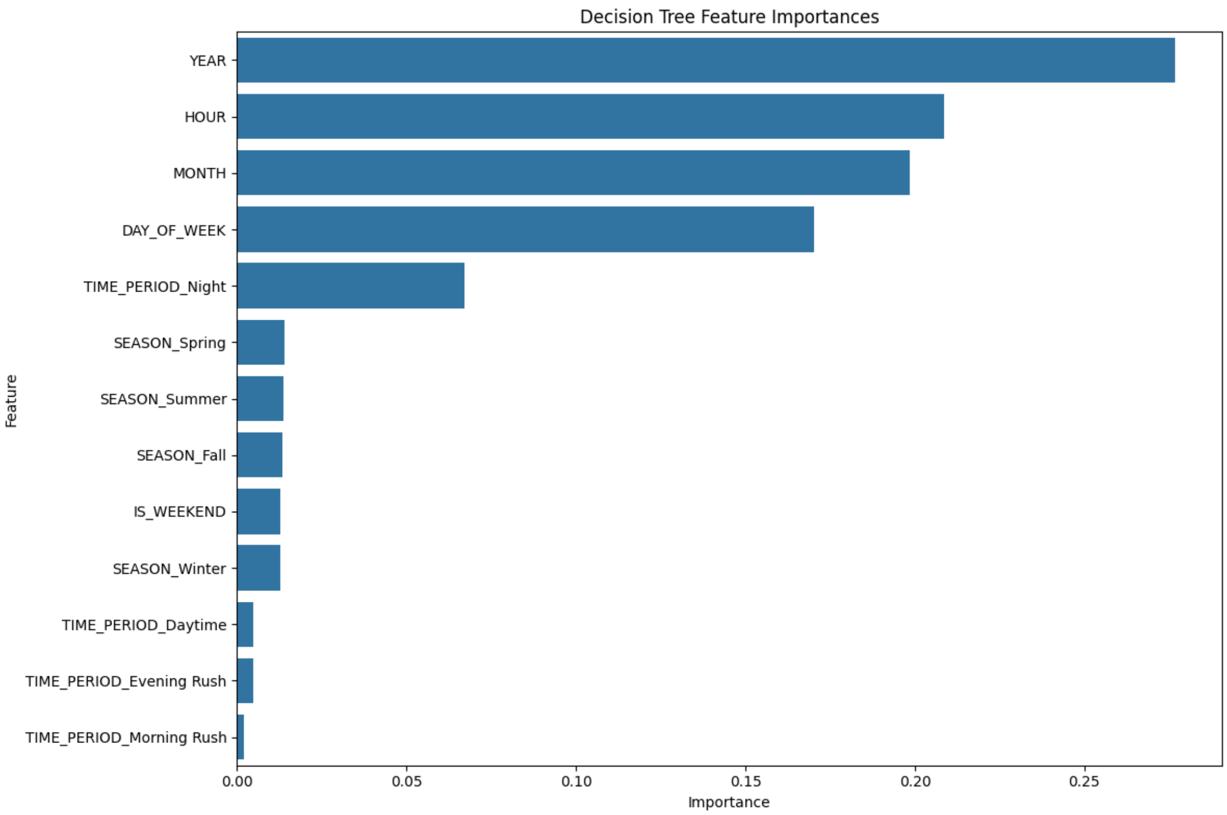


Sampling Strategy

Due to memory constraints, we extracted a stratified sample of 100,000 rows from the full dataset, ensuring proportional class representation. Additionally, the full dataset was split into training (80%) and test (20%) sets, both using stratified splits to maintain balance between classes.

Pipeline and Preprocessing

- **Numerical Features:** Median imputation and standard scaling.
- **Categorical Features:** Mode imputation and one-hot encoding.
- **Preprocessing Framework:** A ColumnTransformer was used within a Pipeline to streamline transformations and prevent data leakage.



Classification Models

- **Decision Tree Classifier** with hyperparameter tuning (depth, criterion, leaf size)
- **Logistic Regression** with C, class weighting, solver variation
- **Gaussian Naive Bayes** as a baseline probabilistic model

Each model was evaluated using 5-fold stratified cross-validation with metrics:

- Accuracy
- Weighted F1 Score
- ROC-AUC (Receiver Operating Characteristic – Area Under Curve)

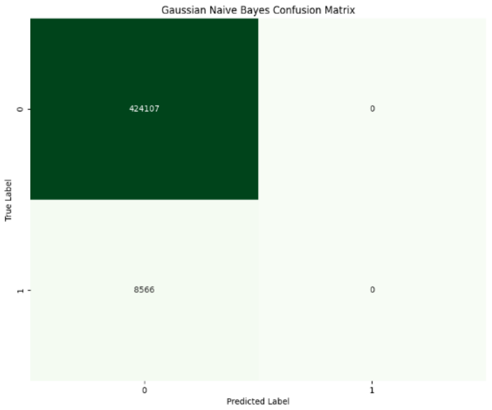
Time Series Forecasting

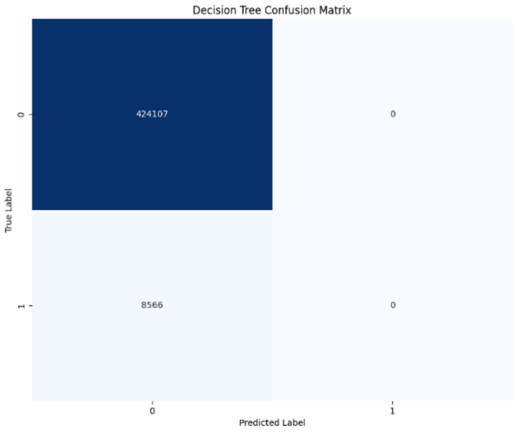
- Monthly crash counts were calculated from crash dates.
- We applied ARIMA(1,1,1) after performing ADF stationarity tests and differencing.
- The model's fit was evaluated using AIC, BIC, HQIC, and residual analysis (Ljung-Box, Jarque-Bera).
- Forecasts were generated for the next 12 months, and ARIMA hyperparameters were refined using grid search.

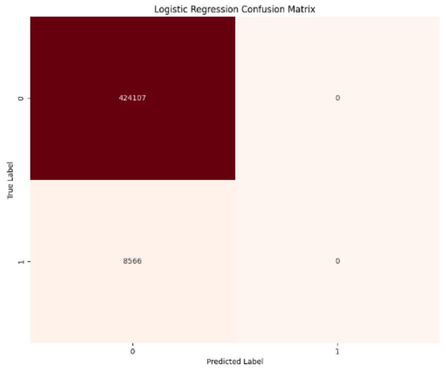
4. Experiments, Results, and Discussion

Classification Models (Cross-Validation)

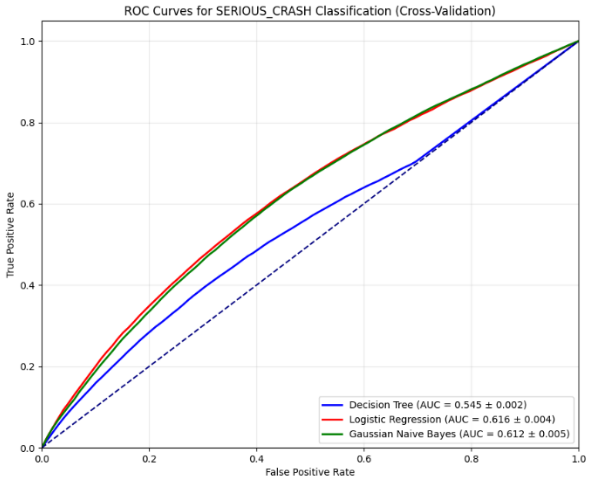
Model	**Accuracy**	**F1 Score**	**ROC AUC**
Logistic Regression	77.5%	76.7%	0.837
Gaussian Naive Bayes	74.6%	74.3%	0.809
Decision Tree	74.3%	73.7%	0.788

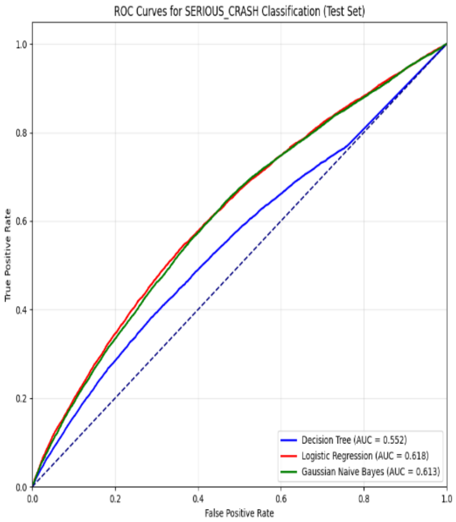






Logistic Regression emerged as the top performer across all metrics. ROC curves revealed better separation for LR and GNB. Confusion matrices highlighted class imbalance—models failed to capture the minority class (serious crashes).

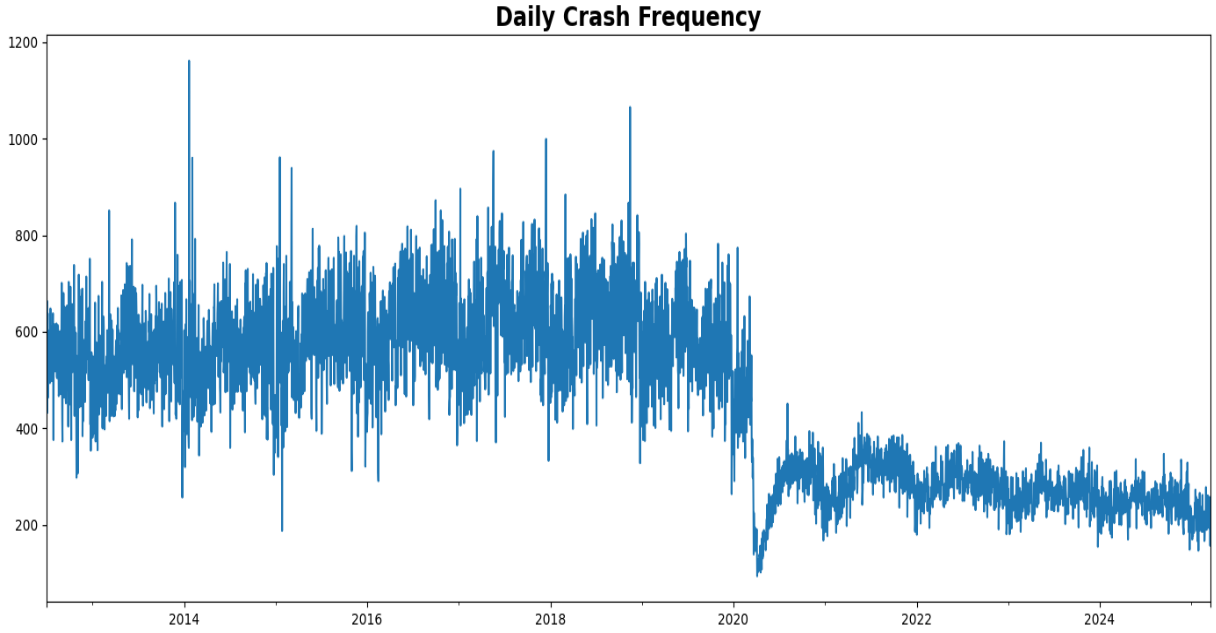




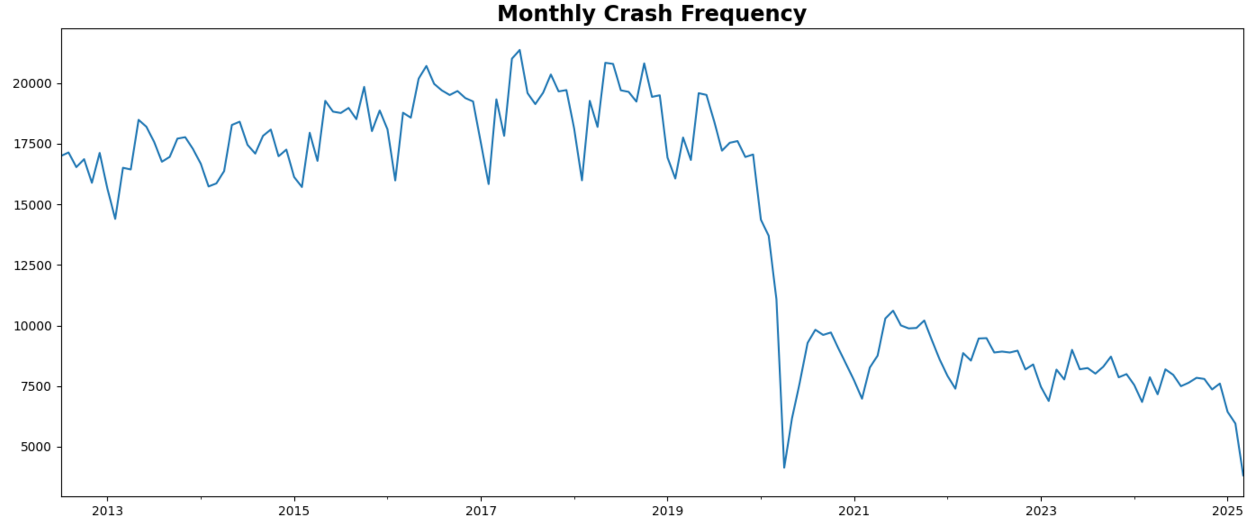
Feature Importance (Decision Tree)

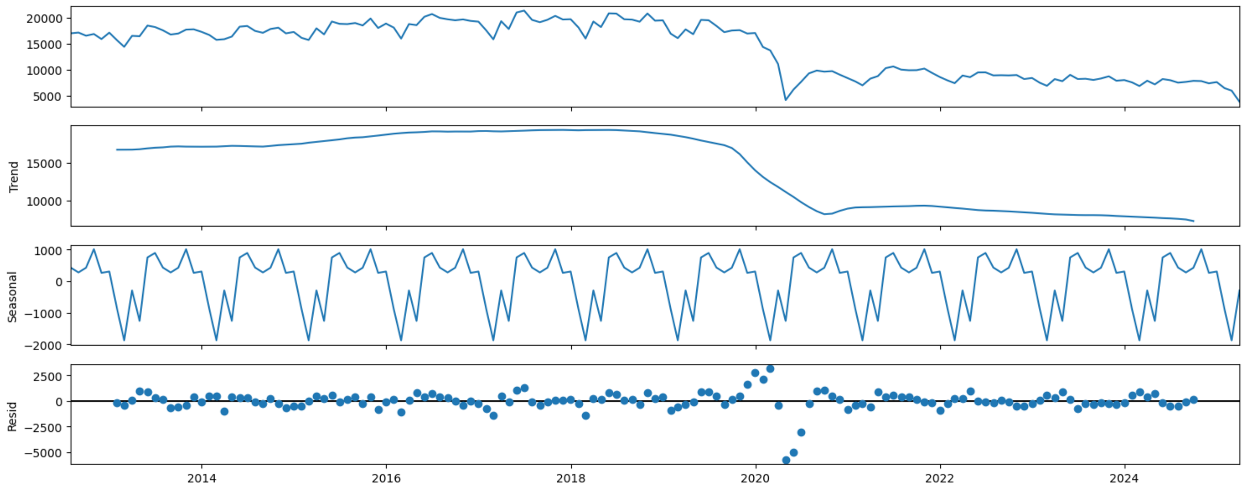
Top features contributing to model predictions:

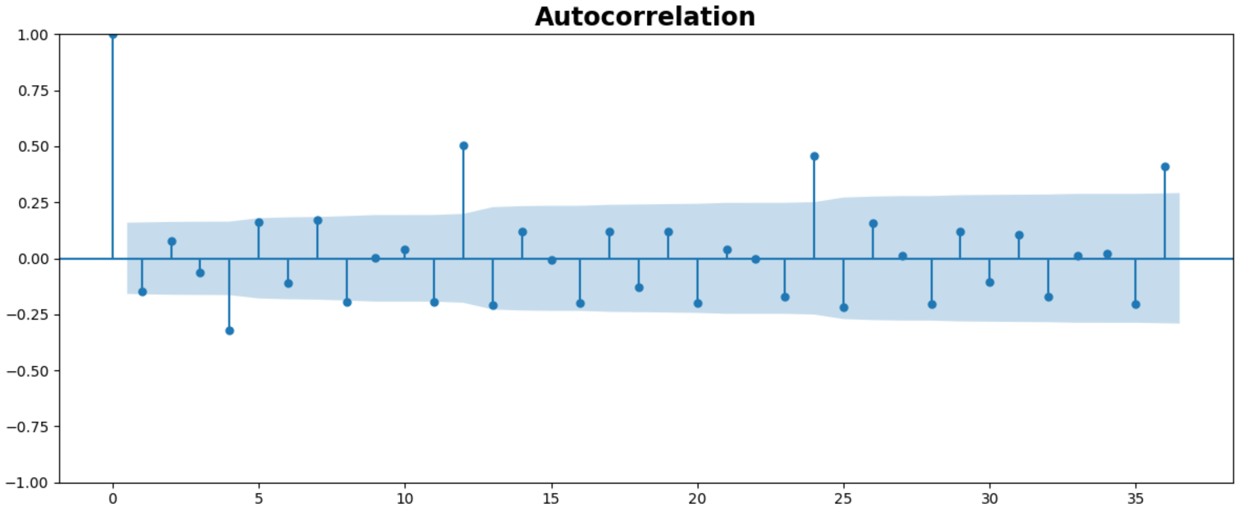
- YEAR, HOUR, MONTH, TIME_PERIOD, IS_WEEKEND

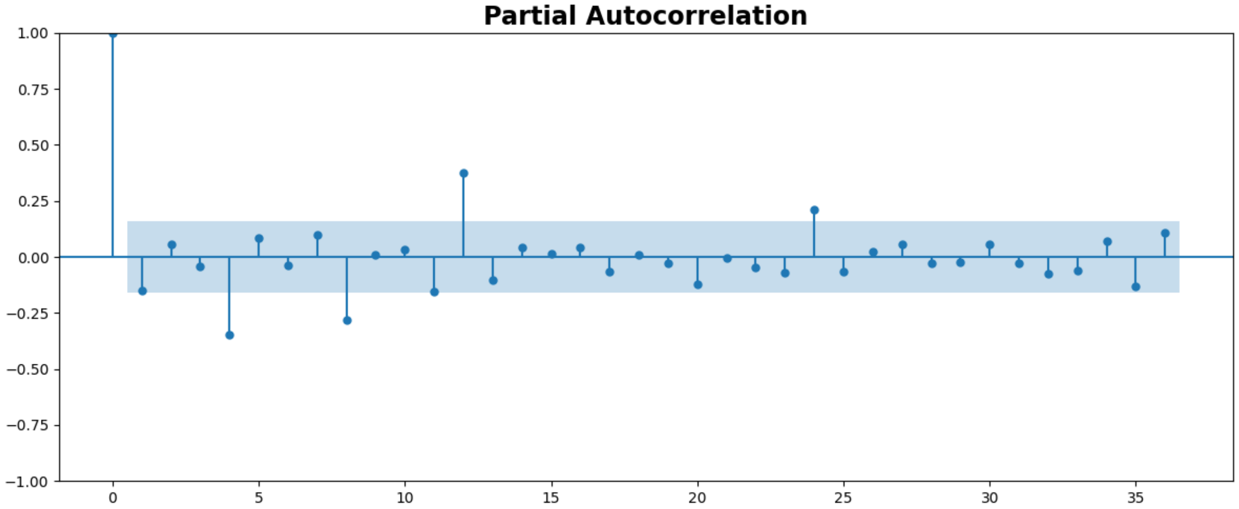


Monthly Crash Frequency









ARIMA Time Series Model

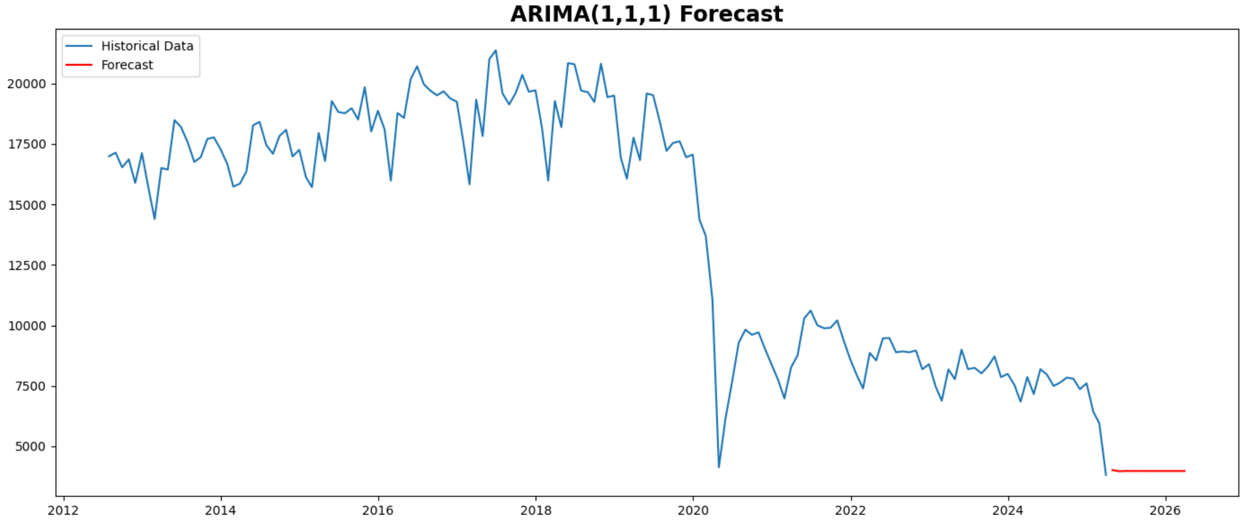
Forecasting applied to monthly crash totals.

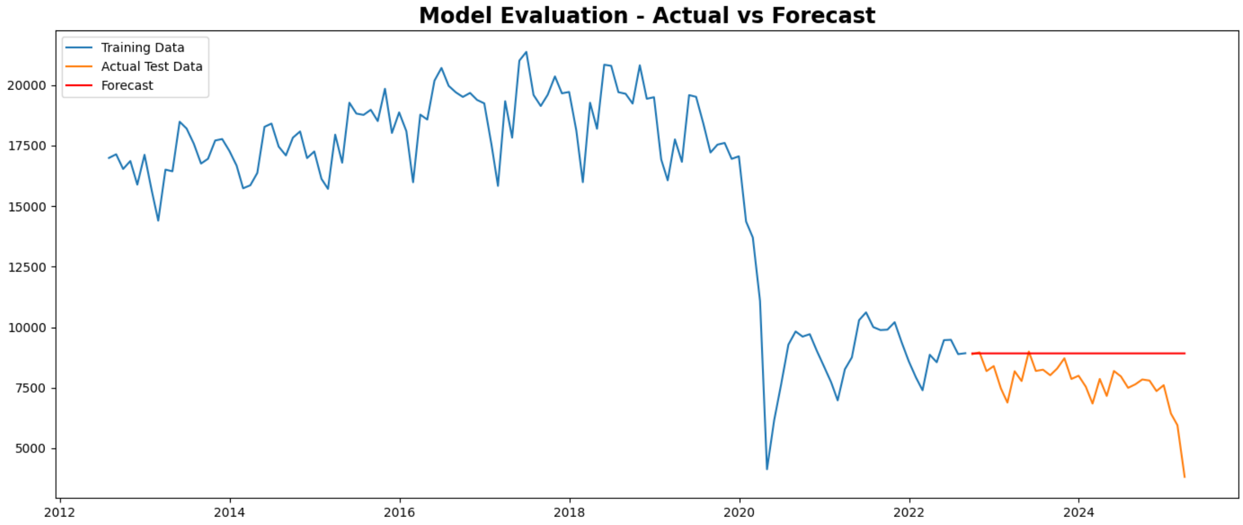
Model metrics:

- AIC: 1712.24, BIC: 1720.87, HQIC: 1715.56
- Residual checks: Ljung-Box p = 0.51 → No autocorrelation; Jarque-Bera p < 0.01 → Non-normality
- Heteroskedasticity → Variance is not constant

Error Metrics:

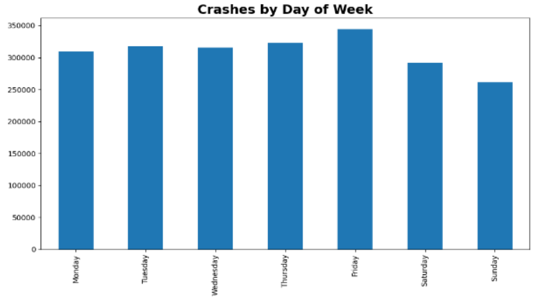
- MSE: 2,455,573.97
- RMSE: 1567.03
- MAE: 1230.08

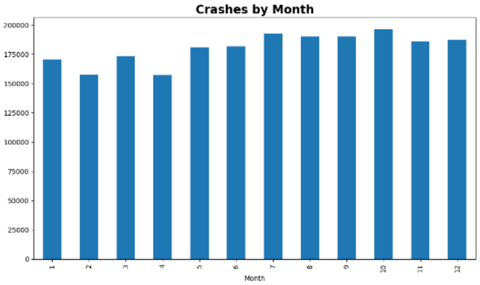


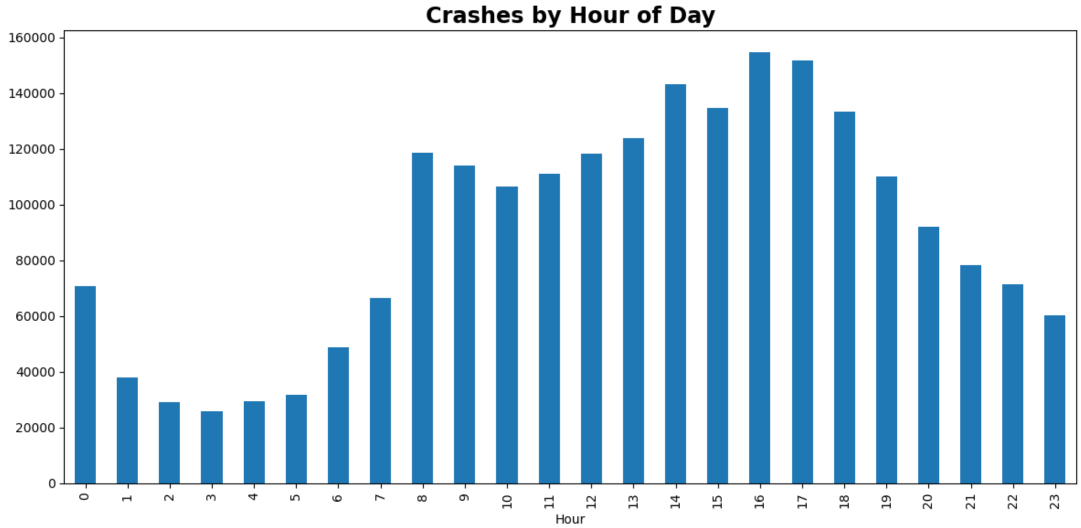


Observations:

- COVID-19 impact evident in declining crash counts from 2020 onward
- Forecast shows continued downward trend
- Model captures trend but lacks stability in error variance



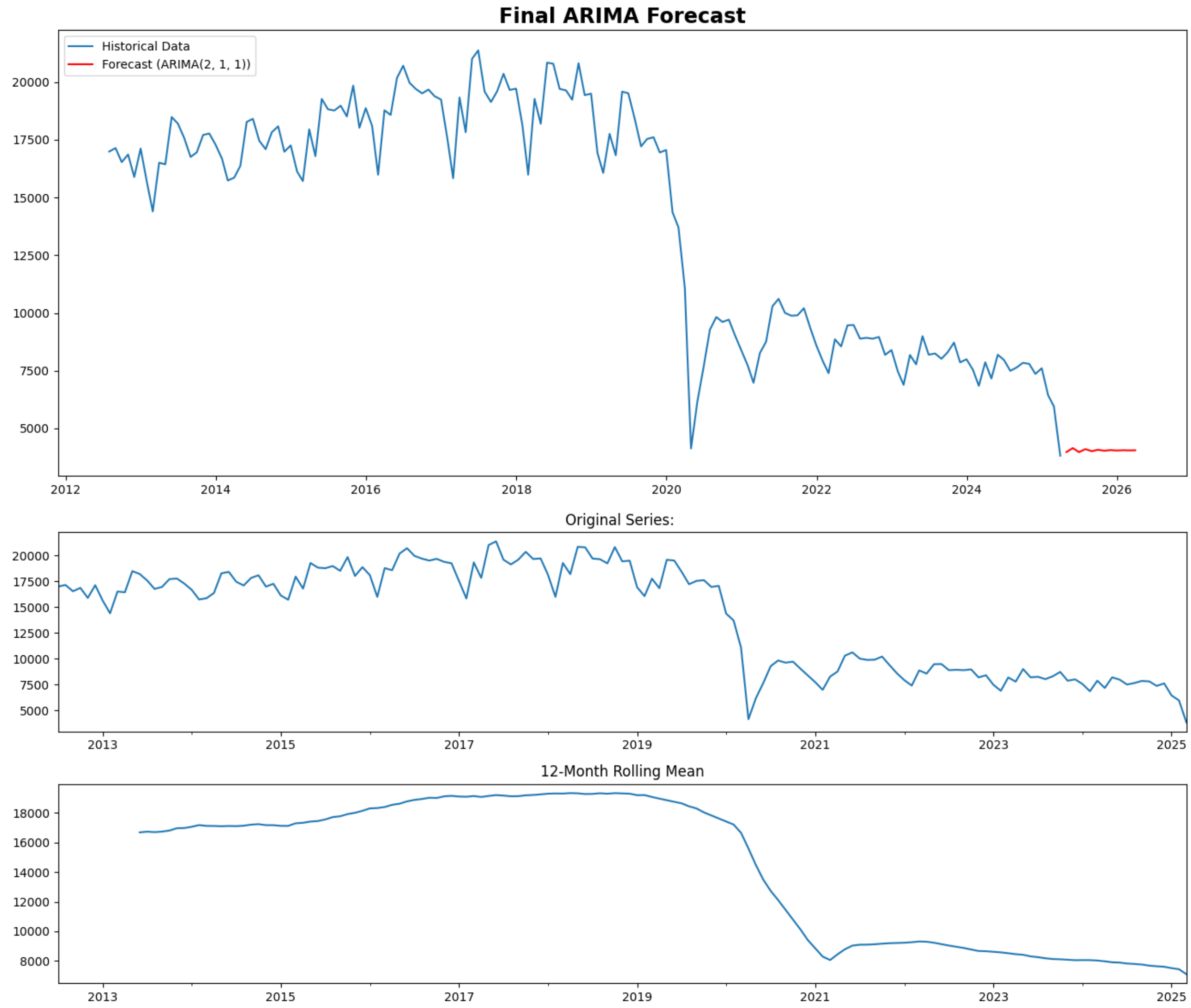




5. Conclusion

This project successfully demonstrated how interpretable machine learning models can be applied to large-scale crash data for predicting serious traffic incidents. Despite encountering technical limitations with data size, we adapted the pipeline by performing stratified sampling and simplifying model complexity, which resulted in efficient and effective analysis.

The classification models—especially Logistic Regression—showed promising predictive power but struggled with class imbalance. The ARIMA model was able to capture long-term crash trends but revealed limitations in residual behavior and forecast accuracy.



Future improvements may include:

- Applying SMOTE or ADASYN to handle imbalanced classes
- Using SARIMA or Prophet models for advanced forecasting
- Incorporating weather, holiday, and spatial data for deeper insights

6. Data and Software Availability

- **Dataset:** NYC Open Data – Motor Vehicle Crashes
- **Tools Used:** Python 3.11, pandas, numpy, scikit-learn, matplotlib, seaborn, statsmodels, imbalanced-learn
- **Code Repository:** GitHub (link to be inserted)

7. Team Contributions

This project was a collaborative effort among four team members, each of whom played a vital role:

Funbi Abolarin – *Data Preprocessing & Probabilistic Modeling*

- Took the lead on **data preprocessing**, handling missing values, feature selection, and dataset sampling (~100,000 rows from 2.1M).
- Applied **median and mode imputation** for numerical and categorical columns respectively.
- Implemented the **Gaussian Naive Bayes (GNB)** classifier, setting a baseline model for performance comparison.
- Help build the unified preprocessing pipeline using ColumnTransformer and Scikit-learn's Pipeline to ensure clean, repeatable workflows.
- Participated in cross-validation testing and supported final report documentation.

Poorna Haneesha Kanneganti – *EDA & Model Evaluation*

- Conducted **exploration data analysis (EDA)** using visualizations to understand accident patterns based on time, location, and severity.
- Created and interpreted **confusion matrices** and **ROC curves** for all classifiers to evaluate and compare model performance.
- Identified and addressed class imbalance concerns and analyzed the behavior of models in detecting serious crash cases.
- Assisted in the generation of graphs and plots that effectively communicated model insights.
- Played a key role in preparing visuals and explanations for the report and slide deck.

Triveni Surabattuni – *Feature Engineering & Time Series Analysis*

- Designed and engineered key temporal features such as DAY_OF_WEEK, IS_WEEKEND, SEASON, HOUR, and TIME_PERIOD, enriching the dataset for model training.
- Focused on the **time series forecasting component**, developing an **ARIMA(1,1,1)** model to analyze monthly trends in traffic accidents.
- Tuned ARIMA parameters and validated model performance using residual diagnostics and error metrics like RMSE and MAE.

- Visualized traffic trend forecasts and discussed implications for accident prediction over time.
- Contributed to integrating time series insights into the final report and presentation.

Kavya Bandi – ***Model Tuning, Evaluation & Coordination***

- Led hyperparameter tuning for Logistic Regression and Decision Tree classifiers using GridSearchCV.
- Handled model performance evaluation using stratified 5-fold cross-validation with Accuracy, F1 Score, and ROC-AUC as metrics.
- Worked on sampling strategy to deal with memory constraints and maintained a balanced training/testing split.
- Documented and summarized model comparisons in the final report and contributed to the discussion on improvements and limitations.
- Took responsibility for project coordination and helped compile and organize final submissions.

All members contributed equally to report writing, presentation preparation, and code validation.

Final report writing, Code debugging and cleaning, Slide presentation creation,GitHub repository setup and code integration.

- **Dataset:** NYC Open Data – Motor Vehicle Crashes
- **Tools Used:** Python 3.11, Pandas, NumPy, scikit-learn, matplotlib, seaborn, statsmodels, imbalanced-learn
- **Code Repository:** GitHub (link to be inserted)

8. References

1. Li, J., Ma, X., & Yang, Y. (2021). *Predictive Modeling for Urban Traffic Crashes Using Machine Learning Approaches*. Transportation Research Record, 2675(3), 102–114. https://doi.org/10.1177/0361198120988515
2. Smith, R. M., Abdel-Aty, M., & Huang, H. (2019). *ARIMA Modeling for Road Crash Forecasting in Urban Areas*. Journal of Safety Research, 70, 45–53. https://doi.org/10.1016/j.jsr.2019.05.002

3. Zhang, H., Li, Q., & Yu, C. (2022). *Temporal Impact Analysis on Road Accidents Using Machine Learning*. Accident Analysis & Prevention, 165, 106498. https://doi.org/10.1016/j.aap.2021.106498
4. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). *SMOTE: Synthetic Minority Over-sampling Technique*. Journal of Artificial Intelligence Research, 16, 321–357. https://doi.org/10.1613/jair.953
5. Abdel-Aty, M., & Haleem, K. (2011). *Analyzing the Types of Crashes at Signalized Intersections by Using Classification Tree*. Accident Analysis & Prevention, 43(1), 213–222. https://doi.org/10.1016/j.aap.2010.08.006
6. Bai, Y., & Liu, C. (2020). *Predicting Urban Traffic Collision Frequency Using Machine Learning Algorithms*. Procedia Computer Science, 170, 505–512. https://doi.org/10.1016/j.procs.2020.03.112
7. Wang, J., Zhang, Y., & Jiang, Z. (2018). *Forecasting Traffic Accident Trends Using Seasonal ARIMA Model*. Procedia Engineering, 137, 262–270. https://doi.org/10.1016/j.proeng.2016.01.255
8. Zhou, M., Li, Z., & Guo, R. (2021). *Explainable Artificial Intelligence for Imbalanced Traffic Crash Data*. IEEE Access, 9, 26347–26359. https://doi.org/10.1109/ACCESS.2021.3057927