Project Report: Predicting Vehicle Collision Causes

1. Overview of Methods and Setup

This project analyzes motor vehicle collision data to predict the primary cause of a crash. The methods and setup are based on the following components:

Dataset

- **Source:** The project utilizes the "Motor Vehicle Collisions - Crashes" dataset, sourced from the U.S. Government open data portal.
- **Temporal Scope:** While the original dataset spans a wider range, this analysis is specifically filtered to focus on collisions occurring in the years **2022, 2023, and 2024**.
- **Data Cleaning:** Initial preprocessing involves converting CRASH DATE to datetime objects, filling numerical NaN values with 0, and filling categorical NaN values with the string "Unknown".

Features and Target Variable

- **Target Variable:** The primary objective is a multi-class classification task to predict the CONTRIBUTING FACTOR VEHICLE 1.
- **Target Classes:** To create a focused and manageable classification problem, the dataset is filtered to include only the **top 7 most frequent contributing factors**:
    1. Driver Inattention/Distraction
    2. Failure to Yield Right-of-Way
    3. Following Too Closely
    4. Backing Unsafely
    5. Other Vehicular
    6. Passing or Lane Usage Improper
    7. Passing Too Closely
- **Predictor Features:** The model uses all other available columns as features. Key predictors include:
    - **Temporal Features:** CRASH DATE and CRASH TIME (used to derive CRASH_HOUR, CRASH_DAY, and CRASH_MONTH).
    - **Spatial Features:** BOROUGH, LATITUDE, LONGITUDE, ON STREET NAME, and CROSS STREET NAME.
    - **Vehicle & Occupant Features:** All other categorical and numerical columns describing the vehicles and persons involved (e.g., vehicle type, number of persons injured/killed).

Algorithms and Methodology

1. **Preprocessing:** All categorical features (e.g., BOROUGH, ON STREET NAME) are encoded using sklearn.preprocessing.LabelEncoder. Datetime columns are converted to integer representations to be compatible with the model.
2. **Class Imbalance Handling:** The filtered dataset remains highly imbalanced. A combined sampling strategy is employed on the training data:
    - **Random Under-Sampling:** The majority class ("Driver Inattention/Distraction") is significantly down-sampled to match the mean count of all classes.
    - **Random Over-Sampling:** All minority classes are then up-sampled to a target count slightly above the mean, creating a more balanced dataset for training.
3. **Modeling:** The core prediction algorithm is a **LightGBM (LGBM) Classifier** (lgb.train). This gradient-boosting model is chosen for its high performance, efficiency with large datasets, and ability to handle categorical features.
4. **Model Explainability:** The project's goal includes providing interpretable insights. This is achieved in the analysis phase by using the model's built-in feature importance (lgb.plot_importance(gbm, importance_type='gain')) to identify which features (like CRASH_HOUR or BOROUGH) are the most influential in predicting collision causes.

2. Updates and Adjustments Since Proposal

Since the initial proposal, several key adjustments have been made to refine the project's scope and address practical challenges discovered during exploratory data analysis and implementation.

- **Pivot from Multi-Label to Multi-Class Classification:** The original proposal aimed to treat the problem as a **multi-label classification** task, where each crash could have multiple simultaneous causes (using CONTRIBUTING FACTOR VEHICLE 1 through 5). However, initial analysis (as noted in the nyc.py file) revealed that the secondary contributing factor columns (CONTRIBUTING FACTOR VEHICLE 2-5) are extremely sparse, with most records having null values.
    - **Adjustment:** To create a more robust and solvable problem, the project has pivoted to a **multi-class classification** task. This focuses solely on predicting the primary CONTRIBUTING FACTOR VEHICLE 1 and filters the dataset to the **top 7 most frequent causes** to ensure sufficient data for each class.
- **Adjustment to Data and Temporal Scope:** The proposal originally suggested training on 2022-2024 data and testing on 2025 data.
    - **Adjustment:** The current implementation uses data from **2022, 2023, and 2024 for both training and testing** (via a standard 80/20 train_test_split). The 2025 data was excluded from this phase, as it is not yet complete for the full year (as of November 2025), which could lead to a skewed or unrepresentative test set.
- **Refinement of Model Scope:** While the proposal listed several potential algorithms (XGBoost, Classifier Chains, Neural Networks), the initial implementation has **focused on the LightGBM (LGBM) Classifier**. This allows for deep tuning and analysis of one high-performing model before expanding to others.

- **Refinement of Evaluation Metrics:** The proposal listed a wide range of multi-label metrics (e.g., Hamming Loss, Subset Accuracy).
    - **Adjustment:** Consistent with the pivot to a multi-class problem, the primary evaluation metrics have been refined to **Weighted F1-score and Macro F1-score**. These are more informative for an imbalanced multi-class problem than simple accuracy.
- **Status of Secondary Tasks (Causal & XAI):** The proposed Causal Inference (PSM, DiD) and advanced XAI (LIME/SHAP) tasks have not yet been implemented. These remain key components of the project's future work, pending the finalization of the core predictive model. The current model explainability is limited to LGBM's built-in feature importance (gain).

3. Experimental Results

The experimental process involved iterating from a complex, broad model to a more focused and higher-performing one.

Initial Experiment: 14 Grouped Classes

An initial experiment was conducted by grouping the 58 unique contributing factors into 14 broader categories (e.g., "Speed/Following Issues," "Lane/Passing Issues"). This approach, while comprehensive, yielded poor predictive performance. Using XGBoost, this 14-class model achieved:

- **Weighted F1-Score:** 0.1509
- **Balanced Accuracy:** 0.2498

The per-class performance was very low, with most classes having F1 scores between 0.11 and 0.25. This result confirmed the need to pivot (as described in Section 2) to a more focused problem: predicting the most common, well-defined primary causes.

Main Experiment: Top 7-Class Model with Resampling

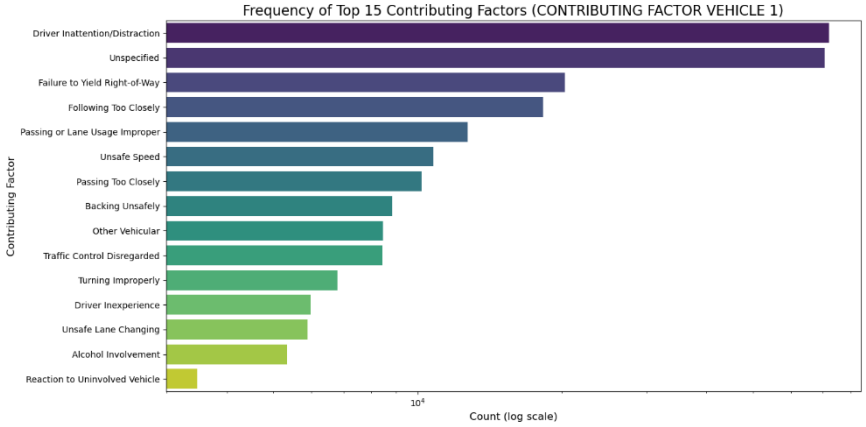
The primary experiment focused on the top 7 most frequent classes. The key to this model's success was addressing the severe class imbalance using a combined under-sampling and over-sampling strategy.

Table 1: Training Class Distribution Before and After Resampling

Class	Original Count	% of Total (Original)	Count After Resampling	% of Total (Resampled)
Driver Inattention/Distraction	57,729	47.8%	17,249	12.3%
Failure to Yield Right-of-Way	16,234	13.4%	20,698	14.8%
Following Too Closely	14,616	12.1%	20,698	14.8%
Passing or Lane Usage Improper	10,182	8.4%	20,698	14.8%

Passing Too Closely	8,156	6.8%	20,698	14.8%
Backing Unsafely	7,071	5.9%	20,698	14.8%
Other Vehicular	6,760	5.6%	20,698	14.8%
Total	**120,748**	**100.0%**	**140,437**	**100.0%**

This balanced dataset allowed the LightGBM model to learn the patterns of minority classes effectively.



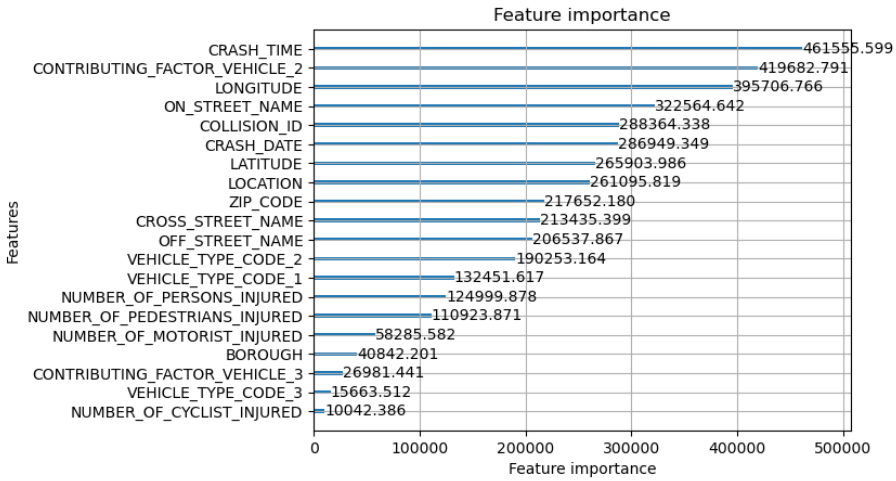
Overall Model Performance

The final LightGBM model, trained on the resampled data, achieved strong aggregate results on the held-out validation set.

```
[RESULT] Weighted F1 : 0.786485695395578
[RESULT] Macro F1    : 0.7780327567210596
[RESULT] Best iter   : 2846
```

Feature Importance

The model's "gain" metric identifies which features were most valuable for splitting data and improving prediction purity. Geographic, temporal, and secondary vehicle information were all highly influential.



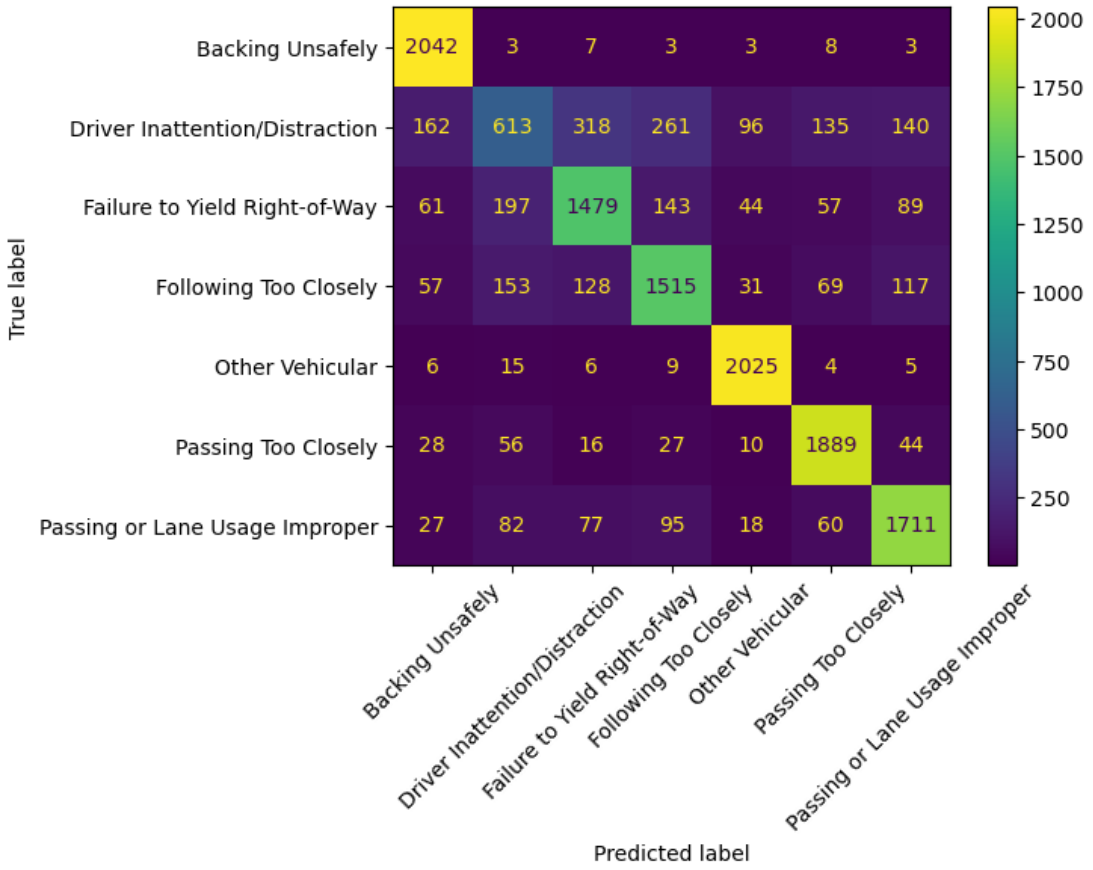
Per-Class Performance

A deeper analysis of the model's performance on each of the 7 classes reveals its specific strengths and weaknesses.

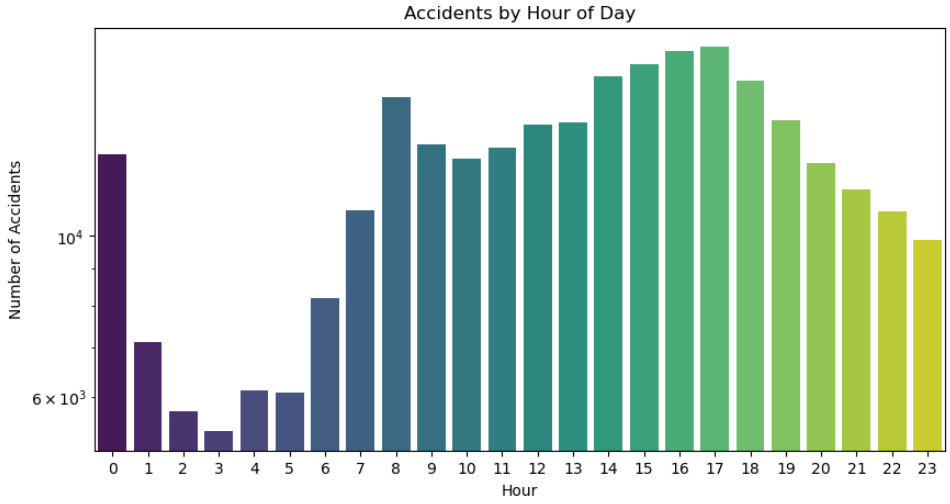
Table 2: Per-Class Validation Accuracy

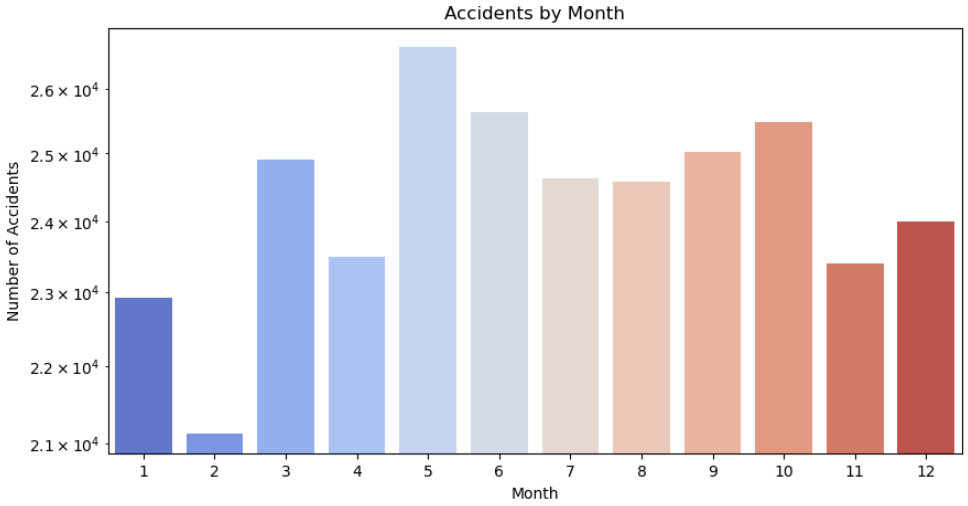
Class	Accuracy
Backing Unsafely	98.7%
Other Vehicular	97.8%
Passing Too Closely	91.3%
Passing or Lane Usage Improper	82.7%
Following Too Closely	73.2%
Failure to Yield Right-of-Way	71.4%
Driver Inattention/Distraction	**35.5%**

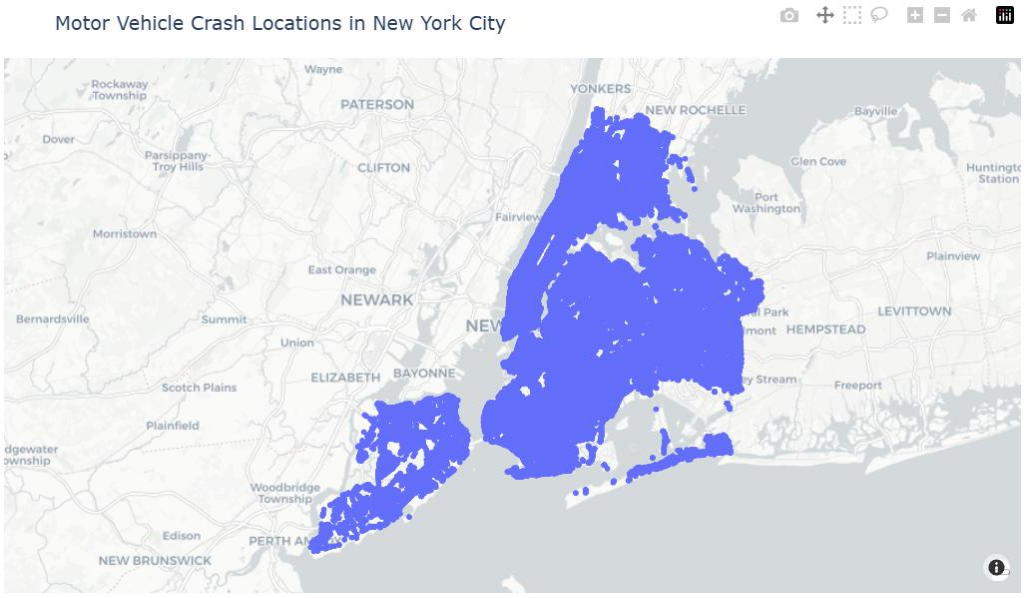
The model's performance is further detailed in the confusion matrix, which visualizes the correct and incorrect predictions for the validation set.



Other meaningful graphs from dataset:







4. Analysis and Discussion

Interpretation of Results

The experimental results demonstrate that the pivot from a 14-class model (F1 0.15) to a focused 7-class model with resampling (F1 0.79) was a successful strategy. The final LightGBM model is highly effective and meets the project's initial objective of building a robust predictive tool, achieving a **Weighted F1-Score of 0.7865**.

The feature importance plot provides a key insight: **context is a major predictor of cause**. The model relies heavily on spatio-temporal features (CRASH_TIME, LONGITUDE, LATITUDE, ON_STREET_NAME) to make its predictions. This confirms the hypothesis that *when* and *where* a collision occurs is deeply correlated with *why* it occurred. The high importance of CONTRIBUTING_FACTOR_VEHICLE_2 is also a significant finding, suggesting that even sparse data, when available, provides a powerful signal that future models should leverage.

Challenges and Limitations

The primary challenge, clearly visible in the per-class accuracy (Table 2) and the confusion matrix, is the model's significant difficulty in distinguishing between a set of related "driver error" classes.

- **High Performance on Distinct Actions:** The model excels at identifying classes with a clear physical signature. "Backing Unsafely" (98.7% acc) and "Other Vehicular" (97.8% acc) are distinct, unambiguous events, and the model predicts them with near-perfect accuracy.
- **Low Performance on Ambiguous Actions:** The model performs very poorly on "Driver Inattention/Distraction" (35.5% acc). The confusion matrix shows these "Inattention" crashes are most often misclassified as "Failure to Yield Right-of-Way" or "Following Too Closely."

This confusion is likely a result of ambiguity in the real-world event and, by extension, the data labels. An officer arriving at a rear-end collision may have to make a judgment call between "Following Too Closely" and "Driver Inattention," for which the available data (time, location, vehicle type) may be identical. The model is struggling with the same ambiguity, indicating that the current features are insufficient to separate these nuanced "driver error" classes.

Future Work and Next Steps

The results suggest a clear path forward, focusing on the main limitations identified.

1. **Address Class Ambiguity:** The immediate next step is to address the confusion between the key "driver error" classes. This will involve **deeper exploratory data analysis and feature engineering** to find signals that can differentiate "Inattention" from "Following Too Closely." This may include analyzing street-level data, weather information, or creating interaction features (e.g., CRASH_HOUR + BOROUGH).
2. **Iterative Model Expansion:** Based on the success of the 7-class model, we plan to carefully **expand the model to include more classes**, moving back toward the original 14 categories. This will be done incrementally, ensuring that the inclusion of new classes (like "Unsafe Speed") does not degrade the performance on the classes already being predicted well. The goal is to **increase the model's scope while maintaining or increasing the overall performance metrics.**

3. **Implement Proposed Secondary Tasks:** With a stable predictive model, we can now proceed with the project's original secondary tasks. This includes implementing advanced **XAI techniques (LIME/SHAP)** to get local, instance-level explanations (beyond the global feature importance) and performing the **Causal Inference analysis** (e.g., Propensity Score Matching) to quantify the impact of environmental factors like lighting and weather, as outlined in the proposal.

5. GitHub Repository

https://github.com/utki007/traffic-accident-ml-classifier