**School of Information Technologies and Engineering, ADA University**

**CSCI4734 – Machine Learning**

**Fall 2024**

**Course Project Report**

# Team 8

|  |  |  |
| --- | --- | --- |
|  | Full Name | Student ID |
| Member 1 | Ali Ahmadli | 14270 |
| Member 2 | Bahruz Gurbanli | 14129 |
| Member 3 | Jeyla Mirzayeva | 16036 |

# Project

## Problem formulation

Formulate the problem. What type of task is it? (50-100 words)

In this project, we are using the Chicago Car Crash Dataset, and following new method is proposed: performing a supervised learning classification to find out the likelihood of car crashes and their results in some circumstances. Besides climate or time-based approaches, our technique handles interactions and multiple characteristics such as weather, road conditions, time of day, longitude and latitude to identify “injury or no injury” situations. To this end, this strategy helps to decrease traffic mortality overall seeks to enhance precision of their predictions and provide valuable information for interventions.

## Discussion of related works (optional)

What has previously been done by others on this topic? (50-100 words)

If we go through available implementations on this dataset, there can be found few analysis/applications in Kaggle platform. Although the work previously has been done covers EDA, features engineering, and predictions of some features, mostly they seem to be too specific and narrow in terms of analysis and implementation of research on it. Thus, the dataset remains underexplored according to the current time (17 December 2024).

## EDA and data preprocessing

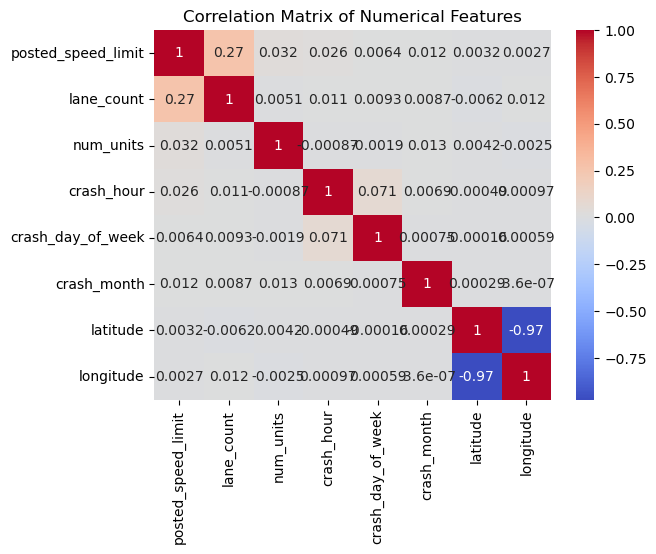
Describe your EDA and data preprocessing steps. Justify your choices. Include any figures and graphs.

Our project encompasses detailed EDA & preprocessing in its implementation. Data is preprocessed by starting from basic things such as renaming columns, removal of unnecessary data and also includes dataset-specific steps such as fixing categorical and numerical features, data cleaning, target imbalance handling and more. To be more specific:

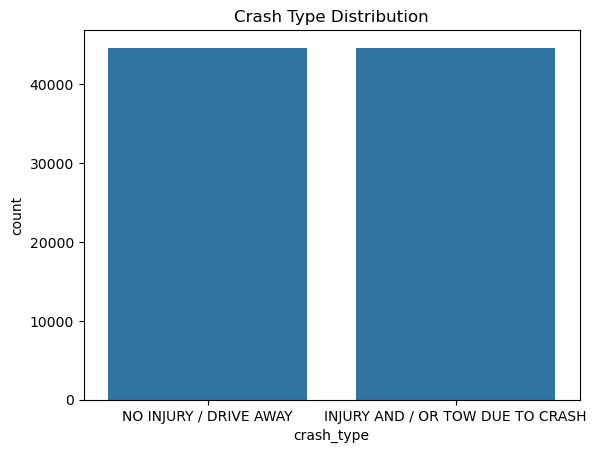
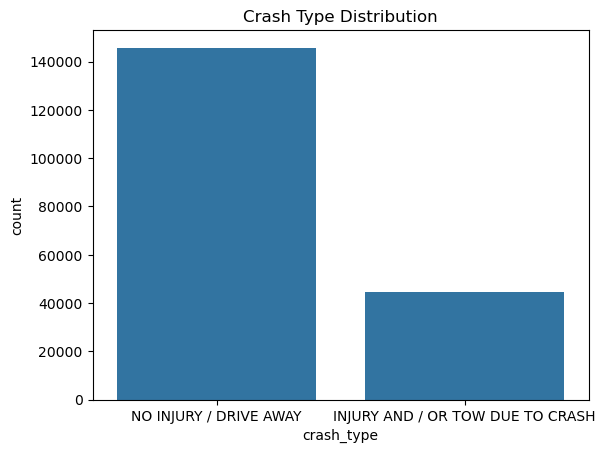
* Some columns were dropped as they were redundant, irrelevant, or added unnecessary complexity: crash\_record\_id (unique identifier), road\_number, street\_number, street\_direction, street\_name, location, and beat\_of\_occurrence (redundant due to latitude and longitude); work\_zone, work\_zone\_type, and workers\_present (too specific and not significant for general crash forecasting); report\_type, photos\_taken, and detailed injury metrics (results of crashes rather than predictors); dooring (specific crash type with little value); and crash\_date\_estimated, crash\_date\_actual, and date\_police\_notified (redundant with crash\_hour, crash\_day\_of\_week, and crash\_month).
* Categorical features standardized by converting to lower case and removing rare categories (less than 1000 occurrences) were grouped as OTHER to prevent overfitting
* Missing values were replaced with UNKNOWN to ensure completeness.
* Numerical features were cleaned by removing invalid rows:
  + posted\_speed\_limit was restricted to values between 0 and 100
  + lane\_count was limited to 1 to 8.

Overall, above mentioned steps ensure a cleaner, simplified, and consistent dataset suitable for modeling for further application.

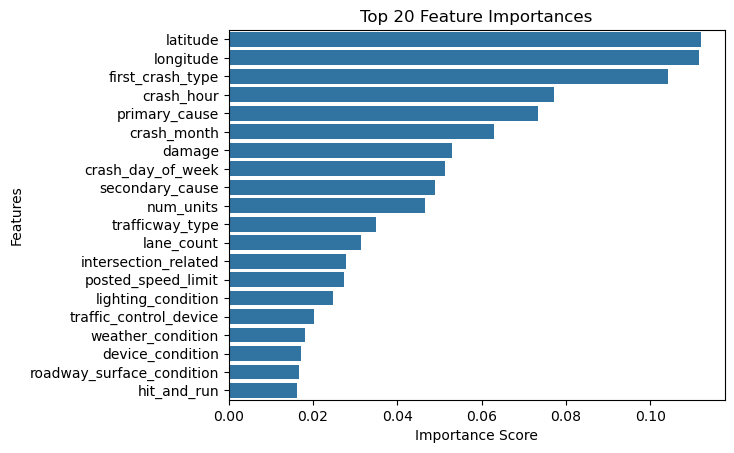
On the other hand, EDA part also includes detailed visualizations of features such as their balance, distribution, count of occurrences, and so on. Here is attached some graphs for visualization of results:



Class Balancing was performed by checking class distribution on the crash\_type column. The NO INJURY / DRIVE AWAY was down sampled to match the INJURY AND / OR TOW DUE TO CRASH.



Feature importances analyzed:



Features with importance score less than 0.01 are dropped: weather\_condition, device\_condition, roadway\_surface\_condition, hit\_and\_run, road\_defect, alignment, not\_right\_of\_way.

(Please check the notebook file containing the source code for detailed visualizations)

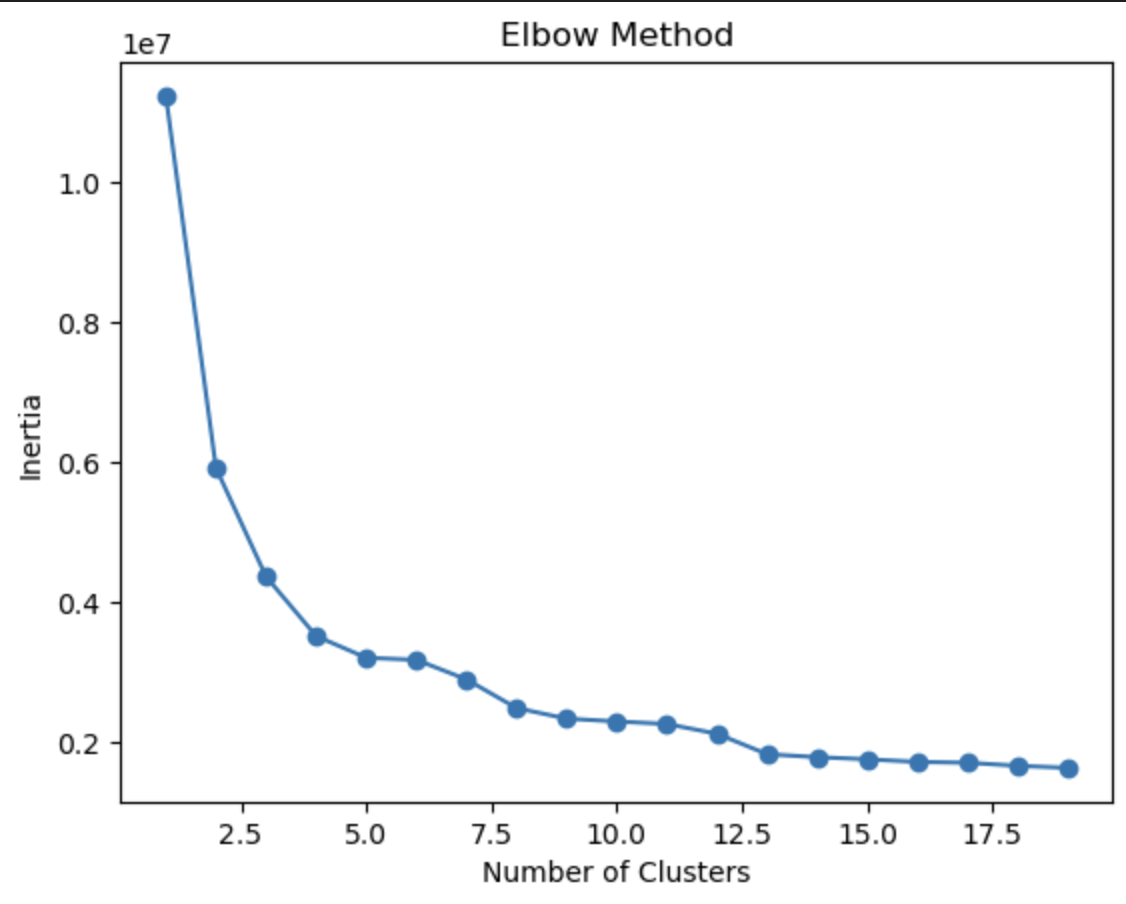
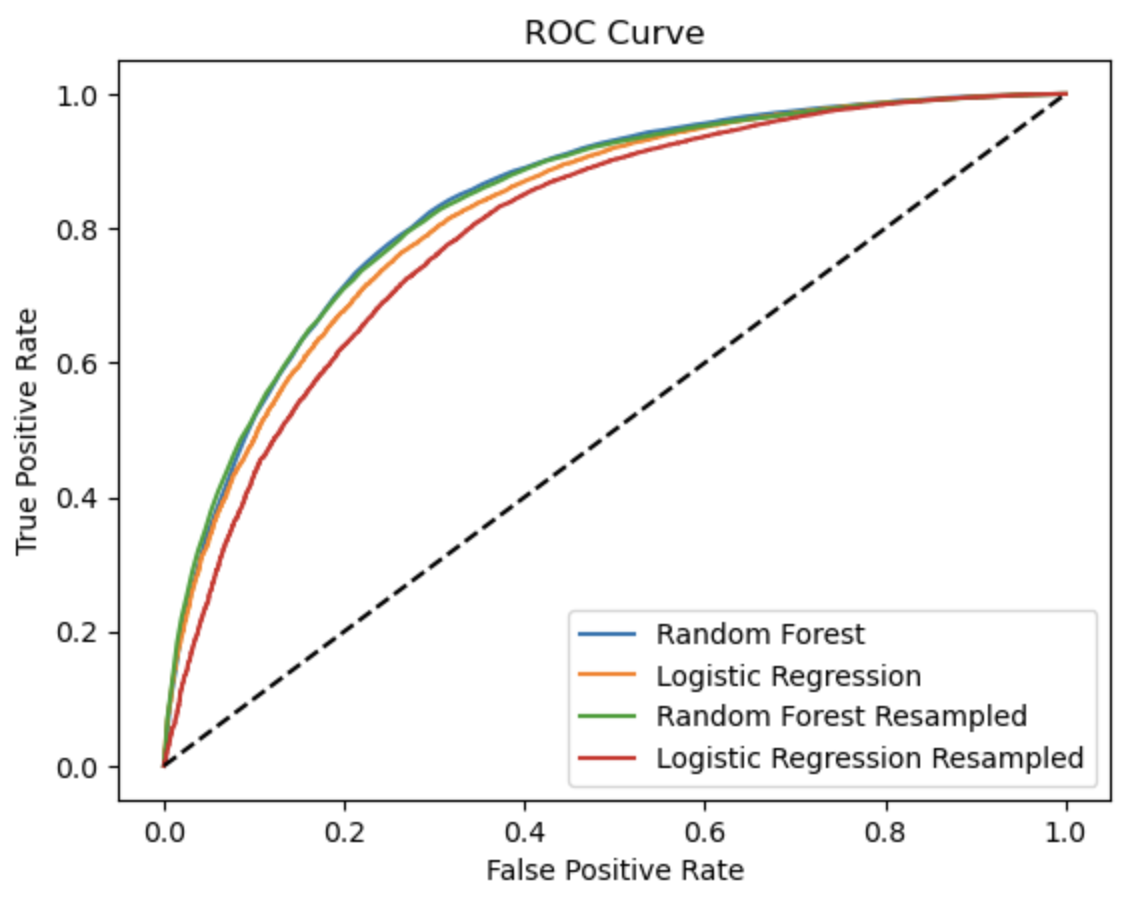
## Modeling

Describe your modeling steps. Which algorithm(s) did you apply and why? What design decisions did you make? How did you evaluate your model? Have you analyzed the errors? Insert any figures.

The modelling methodology is described to predict the target variable. Given the geographical columns in our dataset which have high importance, we did geographic clustering to capture locational patterns. K-means clustering applied to latitude and longitude columns to integrate this spatial context into main models. Multiple numbers of clusters were experienced to find the best number that would successfully capture geographical differences in records. Once the best option was decided using the Elbow method, 3 cluster labels were mapped accordingly. Cluster 1 -> 11 ; Cluster 2 -> 119465 ; Cluster 0 ->70174

Afterwards, two classical machine learning models tested for the prediction task – Random Forest Classifier and Logistic Regression. Random Forest can work with big data and complicated data in general since it uses ensemble learning method and is robust to outliers. In contrast, Logistic regression is chosen as a baseline for its simplicity and effectiveness for linear classification. Nevertheless, it is not so good at generalizing patterns in random data sets but is useful for models to be compared against for finding out how efficient they are.

Initial predictions were made on an unbalanced dataset by Random Forest and Logistic Regression. Our goal was to assess performance of models without any data balancing techniques. Both training sessions include a full set of features. In the next step, imbalances of the dataset were addressed by applying Synthetic Minority Over-sampling Technique (SMOTE).

Random Forest and Logistic Regression models were retrained on the new balanced dataset to assess any performance improvements. In the next stage, RandomizedSearchCV was employed to optimize both models' performances which will efficiently look for best hyperparameters. As a final step, both Random Forest and Logistic Regression models were retrained without the geographical feature for assessing and deciding whether their inclusion was necessary for our model performance. While geographical feature should improve our performance hypothetically based on EDA results, we wanted to see its direct influence on metrics.

For evaluating the performance of our models, several metrics were used: accuracy, ROC curve, and the confusion matrix. Accuracy metric provides us general model performance on how well it does classification, while ROC helps to compare the rate of true positives to false positives. Additionally, confusion matrix helps provide detailed prediction results that can help to define where the models had the most difficulty in classifying.

Eventually, the error analysis encompasses class imbalance, geographic sparsity, and the linear limitations of Logistic Regression which could contribute to misclassifications. SMOTE balancing, geographic clustering, and feature importance analysis helped address these issues, while Random Forest proved to be more robust in minimizing errors.

## Experiments

What experiments did you run? Which MLOps tool(s) did you use? What are some of the advantages/disadvantages of the tool? Insert any figures.

Generally, the experiments focused on baseline modeling, class balancing, geographic clustering, and hyperparameter tuning to optimize model performance. To support the pipeline, MLflow tool was used to keep track of process, enabling comparison, reproducibility, and logging of results. While MLflow significantly enhanced model management, its setup and manual logging require additional effort.

## Discussion of results

How do you interpret the results of the project? Discuss the key points. (100-200 words)

The project’s goal was to classify car crashes into two categories: “No Injury / Drive Away” and “Injury and/or Tow Due to Crash”. To interpret the results, we used several techniques and forms of models, compared them accordingly. Model Choice Matters: While Random Forest’s robustness makes it well-suited for imbalanced and complex datasets, Logistic Regression’s simplicity and considerable comparison chance is discussed. To handle the imbalance SMOTE technique was used to ensure fair performance across classes. As crash density can vary spatially, and clustering analysis can provide actionable insights for safety improvements we consider both cases while training and comparisons.

Random Forest achieved ~83% accuracy and consistently outperformed Logistic Regression (~75% accuracy) in all experiments due to its ability to handle complex and ability on non-linear relationships. Random Forest showed its higher precision and recall for the majority class but moderate misclassification of minority class. Moreover, the models trained and tested after SMOTE balancing. It effectively reduced the class imbalance, improving minority class performance: Before SMOTE, false negatives were looking harmful (injury-related crashes misclassified as non-injury). After SMOTE, minority class recall increased, indicating fewer injury-related crashes were missed. Random Forest accuracy without resampling: 83% and with resampling 82.4% (affected negatively). Logistic Regression accuracy without resampling: 75% and with resampling 78.4% (affected positively). After applying hyperparameter Tuning, Random Forest slightly improved accuracy and recall, Logistic Regression did too nevertheless it could not bridge the gap with Random Forest. (Fitting 2 folds for each of 20 candidates, which totalling 40 fits. Actually, parameters could be increased accordingly if it would not take too long time to train).

