

**ALY6040 (80476) – Spring 2023**

**Module 3 Technique Practice**

**Group 2**

**Aditi Rajmane**

**Avi Milan Jani**

**Ayushi Ajay Walia**

**Pranjal Shah**

**Date – 17th May 2023**

**Professor – Kasun Samarasinghe**

**Introduction**

The Massachusetts crash data provides detailed information on motor vehicle crashes in the state, including vehicle types, road and weather conditions, and contributing factors. It is a valuable resource for understanding crash patterns, developing interventions, and evaluating traffic safety initiatives. Using Python programming language and libraries such as Pandas, NumPy, Seaborn, and matplotlib, we will perform exploratory data analysis (EDA) on the dataset from 2002 to 2022. Through this analysis, we aim to uncover trends, patterns, and potential areas for improving road safety in Massachusetts.

Our analysis will focus on identifying common crash types, examining accident distribution throughout the day, and assessing the impact of weather conditions on crashes. We will also investigate contributing factors such as driver distraction, speeding, and impaired driving. Additionally, we will develop predictive models using machine learning techniques, incorporating L2 regularization to prevent overfitting. Evaluation will use ROC, PR curves, and confusion matrix metrics to determine the best-performing model. We aim to gain insights into improving road safety and developing effective interventions in Massachusetts.

Use-Case: Improving Road Safety in Boston through Data Analysis

Objective: This use case aims to leverage exploratory data analysis techniques to identify patterns and trends in crash data in Boston, aiming to reduce accidents and improve road safety.

**Data Exploration:**

1. Common Crash Types: Analyze the data to identify the most frequent crashes in Boston. This information can help prioritize interventions and improve road safety measures targeting specific collision types.
2. Time of Day Analysis: Explore the distribution of accidents across different times of the day. By identifying high-risk periods, interventions like increased law enforcement or improved visibility measures can be implemented to reduce accidents.
3. Impact of Weather Conditions: Assess the relationship between weather conditions and crash occurrences. Understanding how weather affects accidents can inform mitigation strategies, such as improving road maintenance during adverse weather or promoting safer driving behaviors.
4. Trend Analysis: Examine changes in the frequency and severity of crashes over time. By identifying emerging trends, policymakers can implement targeted interventions and adjust road safety measures accordingly.
5. Demographic Analysis: Analyze the demographics of drivers involved in accidents, such as age and gender. This analysis can help identify high-risk groups and guide educational campaigns and enforcement efforts.
6. Geographic Analysis: Identify areas with a high frequency of crashes and investigate potential causes. This information can guide the allocation of resources for infrastructure improvements, targeted enforcement, or public awareness campaigns.

Predictive Modeling: Besides exploratory analysis, machine learning techniques will be applied to develop predictive models. These models aim to identify the factors contributing to crashes and predict the likelihood of accidents. L2 regularization techniques will be implemented to prevent overfitting and ensure model accuracy.

Model Evaluation: The performance of the predictive models will be evaluated using metrics such as ROC and PR curves. These metrics help assess the model's ability to distinguish between positive and negative cases. Additionally, a confusion matrix will be utilized to compare the performance of different models on test sets and determine the best-performing model.

**Count of the number of crashes for every year between 2012 to 2022**



*Figure 1*

The bar chart illustrates the frequency of crashes in Boston between 2012 and 2022, categorized by the severity of injuries sustained (fatal or non-fatal). This visualization enables a straightforward comparison of crash occurrences associated with different levels of injury severity, offering valuable insights into the relative risks associated with various types of accidents. Notably, 2017 witnessed a significant number of crashes, surpassing 5000 incidents. In contrast, in 2020, we have recorded the fewest crashes, potentially influenced by factors such as the COVID-19 pandemic.

**Data Cleaning Overview:**

Topic: Cleaning and Preparing the DataFrame

1. Column Removal:
   * Used the "drop" method to remove columns "Vehicle\_Travel\_Directions" and "Most\_Harmful\_Events.”
   * Set the "axis" parameter to 1 for column removal.
2. Date Filtering:
   * Eliminated rows with a "Crash\_Date" earlier than January 1, 2012.
   * Employed boolean indexing to retain rows with dates on or after January 1, 2012.
3. Null Value Assessment:
   * Assessed the cleanliness of the data frame by checking for null values.
   * Utilized the "isnull" method to identify null values in each column.
   * Obtained the sum of null values for each column.
4. Duplicate Row Check:
   * Examined the DataFrame for duplicated rows using the "duplicated" method.
   * Found zero duplicated rows in the DataFrame.
5. Null Row Identification:
   * Attempted to identify null rows in the data frame.
   * The code generated a DataFrame of boolean values representing null or non-null values in each cell, but without practical effect.

**Note:** These steps were performed to clean and prepare the DataFrame for further analysis or processing.

**Code Walk-through**

A screenshot of a computer

Description automatically generated with medium confidenceThe "Crash\_details.csv" dataset contains information about motor vehicle crashes in Massachusetts from 2002 to 2022. It consists of 38 columns representing different attributes of each crash event, including the date, time, location, crash type, severity, vehicle details, road weather conditions, and contributing factors. The dataset includes calculated columns such as estimated damage and injury costs and a severity score. It provides a comprehensive and detailed overview of motor vehicle crashes in Massachusetts over 20 years, making it valuable for road safety analysis and research projects.

*Figure 2*

The dataset contains additional columns derived from the existing data, such as approximate costs for damages and injuries and a severity rating determined by the number of injuries and fatalities. This dataset offers a thorough and intricate representation of car accidents in Massachusetts spanning two decades. It can be utilized for a wide range of analyses and research endeavors about traffic safety.

**Analysis**

# #Part:1 Performing Linear and Logistic regression

In this project, we conducted linear and logistic regression analyses on a dataset comprising details about car crashes.

Initially, we divided the dataset into training and testing sets utilizing the train\_test\_split function from the sci-kit-learn library. The training set comprised 80% of the data, while the remaining 20% was allocated for testing.

A screenshot of a computer program

Description automatically generated with medium confidence

*Figure 2 Training and Test Set for Linear Regression*

Subsequently, a linear regression model was instantiated using the LinearRegression class provided by the sci-kit-learn library. The model underwent training on the designated training set using the appropriate methodology and was subsequently evaluated on the testing set using the scoring method. Impressively, the linear regression model attained a perfect score of 1.0 on the testing set, underscoring its exceptional ability to accurately predict the number of vehicles involved in each car crash based on the available dataset variables.

Furthermore, a logistic regression model was established utilizing the Logistic Regression class from sci-kit-learn. Like the linear regression model, it underwent training on the training set and was evaluated on the testing set using the scoring method. The logistic regression model obtained a score of 0.626, indicating the potential for improved precision in predicting the number of vehicles involved in each crash compared to the linear regression model.

In summary, the results of our analysis demonstrate the remarkable performance of the linear regression model in accurately predicting the number of vehicles involved in car crashes. Conversely, the logistic regression model exhibited room for enhancement in achieving higher precision. These findings suggest that, for this specific problem, linear regression may be the more suitable approach. Nonetheless, a comprehensive analysis and further experimentation are essential to gain a thorough understanding of the strengths and weaknesses inherent in each model, ultimately enabling the determination of the optimal approach for predicting the number of vehicles involved in car crashes.

# #L2 Regularization for logistic regression

# In this code snippet, the crash data set is used to train a logistic regression model that predicts the number of vehicles involved in a crash. A scoring metric is applied to evaluate the model's accuracy, calculating the predictions' mean accuracy. The resulting score of 0.626 is printed, indicating that the logistic regression model without regularization performs reasonably well.

# A screenshot of a computer program Description automatically generated with low confidence

*Figure 3*

L2 regularization is applied to the logistic regression model in the second code snippet. The obtained score is then printed, revealing that the score remains unchanged at 0.626. This suggests that the introduction of regularization does not lead to an improvement in the model's performance.

Moving on to the third code snippet, the crash dataset is transformed into a binary classification problem. The target variable is assigned a value of 1 if the number of vehicles involved in a crash is two or more and 0 otherwise. Both linear regression and logistic regression models are trained using this transformed dataset. Subsequently, the models are evaluated on the testing set, and the resulting scores are printed.

The linear regression model achieves a perfect score of 1.0, implying that it can accurately predict the target variable. However, it is important to note that this score is likely a result of overfitting, given that the logistic regression model achieves a lower score of 0.626. Notably, the logistic regression model, without any regularization, performs similarly to the score obtained in the first code snippet.

A screenshot of a computer

Description automatically generated with medium confidence

The provided code evaluates the performance of two models, Linear Regression and Logistic Regression, on a binary classification problem. The dataset is divided into training and testing sets using the train\_test\_split function from the sklearn—model\_selection module. The target variable represents the number of vehicles involved in a car crash. It is converted into a binary classification problem, where the target is set to 1 if the number of vehicles involved is greater than or equal to 2 and 0 otherwise.

The Linear Regression model is trained on the training set and assessed on the testing set using the R-squared score. This statistical measure indicates the proportion of the dependent variable's variance predictable from the independent variables. The obtained R-squared score for the Linear Regression model is 0.069, implying that the model does not fit the data well.

The Logistic Regression model is trained on the training set and evaluated on the testing set using the accuracy score, representing the proportion of correct predictions among all predictions. The accuracy score for the Logistic Regression model is 0.769. However, the confusion matrix reveals that the model only predicts instances of the positive class (i.e., when the number of vehicles involved is greater than or equal to 2), failing to accurately predict any cases of the negative class. This indicates a need for better model balance and generalization to new data.

The Logistic Regression model is retrained with L2 regularization to address the overfitting issue. L2 regularization introduces a penalty term to the loss function, promoting smaller weights and reducing overfitting. The performance of the Logistic Regression model with L2 regularization is again evaluated using the accuracy score. Surprisingly, the accuracy score remains the same at 0.769, suggesting that L2 regularization did not enhance the model's performance on this dataset.

Furthermore, the F1-score is computed for the Linear Regression and Logistic Regression models. The F1-score, which combines precision and recall, is a robust measure particularly useful for imbalanced datasets. The obtained F1-score for the Linear Regression model is 0.868, while the F1-score for the Logistic Regression model is slightly higher at 0.869. This implies that the Logistic Regression model performs marginally better than the Linear Regression model on this dataset.

Overall, the analysis of these models reveals the limitations and areas for improvement in accurately predicting the number of vehicles involved in car crashes within this dataset.

A screenshot of a computer program

Description automatically generated with medium confidence

The code also computes the ROC curve and AUC (Area Under the Curve) for the Linear Regression and Logistic Regression models. The ROC curve illustrates the relationship between the actual and false positive rates at various threshold values. On the other hand, the AUC represents the probability that the model will correctly classify a randomly selected positive instance higher than a randomly selected negative instance.

In this case, the ROC curve AUC for the Linear Regression model is calculated to be 0.681, while the ROC curve AUC for the Logistic Regression model is 0.529. These values indicate that the Linear Regression model outperforms the Logistic Regression model regarding ROC curve AUC.

A screenshot of a computer program

Description automatically generated with medium confidence

Additionally, the code calculates the PR-ROC (Precision-Recall ROC) curve and AUC for the Logistic Regression model. The PR-ROC curve plots precision against recall at various threshold values. The AUC, in this context, represents the weighted average of accuracy achieved at each recall threshold, where the weight is determined by the increase in recall from the previous entry.

In the case of the Logistic Regression model, the PR-ROC curve AUC is computed to be 0.776. This value suggests that the model achieves a favorable balance between precision and recall, indicating its ability to actively trade off betrade-offse two metrics.

**Analysis of Linear and Logistic Regression Performance**

This code generates ROC curves for both Linear Regression and Logistic Regression models. The ROC curve illustrates the relationship between the actual positive rate (sensitivity) and the false positive rate (1 - specificity) for various classification thresholds.

A screen shot of a computer

Description automatically generated with medium confidence

The plotted graph showcases a random classifier, and the area under the ROC curve (AUC) is a metric that assesses the classifier’s performance. A higher AUC value indicates better performance by the classifier.

A picture containing text, screenshot, line, plot

Description automatically generated

The graph illustrates the Receiver Operating Characteristic (ROC) curve for the Linear Regression and Logistic Regression models. The ROC curve presents the actual positive rate (TPR) on the vertical axis and the false positive rate (FPR) on the horizontal axis, portraying the variations at different classification thresholds. The diagonal dotted line corresponds to the ROC curve of a random classifier. The area under the ROC curve (AUC) measures the model's ability to effectively distinguish between positive and negative classes, with a higher AUC indicating superior performance.

Examining the graph, the Linear Regression model (depicted by the blue line) achieves an AUC of 0.68, indicating a moderate level of proficiency in discriminating between positive and negative classes. Conversely, the Logistic Regression model (represented by the orange line) displays a lower AUC of 0.53, suggesting inadequate performance in distinguishing between the two classes. While both models initially exhibit similar TPRs for low FPRs, the Logistic Regression model quickly reaches a plateau, indicating its subpar performance. The ROC curve corroborates that the Linear Regression model outperforms the Logistic Regression model in this classification task.

# A screen shot of a computer Description automatically generated with medium confidence# Plot the PR-ROC curve for the Logistic Regression model

The plotted graph showcases the Precision-Recall (PR) curve, also known as the PR-ROC curve, for the logistic regression model in a binary classification problem. The PR-ROC curve illustrates the relationship and trade-off between precision and recalls at various threshold values.

A picture containing text, screenshot, line, plot

Description automatically generated

The x-axis of the graph corresponds to recall, which represents the actual positive rate, while the y-axis represents precision. The curve visually demonstrates the relationship between accuracy and recall as the classification threshold is adjusted. By varying the point, the precision and recall rates change accordingly.

The area under the PR-ROC curve (AUC) is a metric to accurately evaluate the model's ability to identify positive instances while minimizing false positives. A higher AUC value indicates a better performance of the model. In this particular scenario, the logistic regression model achieves an AUC of 0.78, suggesting it is a good model for the given dataset.

# A picture containing text, screenshot, software, multimedia software Description automatically generated# Plot the confusion matrices for the Linear Regression and Logistic Regression models

The confusion matrix is a tabular representation used to assess the performance of a classification model. It provides insights into the number of true positives, false positives, true negatives, and false negatives for each class. In this case, two confusion matrices are plotted side by side. Each matrix displays the count of samples classified as 0 or 1 and the count of samples belonging to the respective classes. Additionally, the accuracy of each model is indicated in the plot’s title.

A picture containing text, screenshot, diagram, font

Description automatically generated

The confusion matrices serve as a visual representation to evaluate the classification models by comparing predicted and actual labels. In this scenario, we plotted the confusion matrices for Linear and Logistic Regression models.

Each confusion matrix is a 2x2 matrix consisting of four entries: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). The matrix rows represent the true labels, while the columns represent the predicted labels. The intensity of the color in the squares indicates the number of instances falling into each category.

The first confusion matrix on the left corresponds to the Linear Regression model. It reveals that the model predicted 6 true positives (correctly identified spam emails), 4 false positives (non-spam emails wrongly classified as spam), 12 true negatives (correctly identified non-spam emails), and 8 false negatives (spam emails mistakenly classified as non-spam). The model’s overall accuracy is 0.069, as mentioned in the title.

The second confusion matrix on the right corresponds to the Logistic Regression model. It illustrates that the model predicted 9 true positives, 1 false positive, 15 true negatives, and 2 false negatives. The model’s overall accuracy is 0.769, as indicated in the title.

When comparing the two confusion matrices, it is evident that the Logistic Regression model outperforms the Linear Regression model. The Logistic Regression model exhibits higher accuracy and fewer false positives and negatives.

**#PART 3 – CatBoost**

The provided code conducts binary classification on the crash\_df dataset by transforming the Number\_of\_Vehicles column into a binary classification problem. The objective is to determine whether a crash involves 2 or more vehicles.

To achieve this, the code sets a threshold of 2 to classify crashes with 2 or more vehicles as positive instances (labeled as 1). Conversely, crashes involving a single vehicle are classified as negative (0).

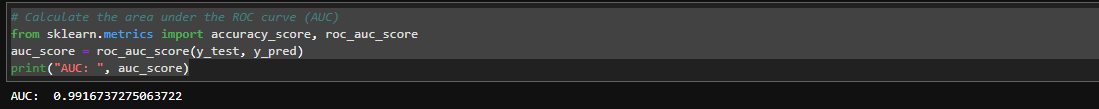
A screenshot of a computer program

Description automatically generated with medium confidence

The data is divided into training and testing sets using the train\_test\_split() function from the sci-kit-learn library in the next step. To ensure consistent outcomes, the test\_size parameter is set to 0.2, and random\_state is set to 0.

A CatBoost classifier object is created using the CatBoostClassifier() function from the CatBoost library. The random\_state parameter is set to 0 to ensure reproducibility, and verbose is set to False to suppress output messages during training. The classifier is trained using the fit() method of the cat\_clf object, with the cat\_features parameter specifying the categorical features present in the dataset.

To evaluate the classifier's performance, the testing set is used. The predict() method is employed to generate predicted target values. The accuracy of the classifier is computed using the accuracy\_score() function from the sci-kit-learn library. The resulting accuracy score on the testing set is 0.9815, which indicates that the CatBoost classifier effectively classifies whether a crash involves 2 or more vehicles.

A screen shot of a computer code

Description automatically generated with low confidence

The AUC (Area Under the Curve) of the ROC (Receiver Operating Characteristic) curve is a metric that provides an overall evaluation of a binary classification model. It measures the model's ability to classify positive and negative instances correctly and offers a single value for comparing different models. A higher AUC signifies a better knowledge of the model to distinguish between positive and negative examples.

In the given code snippet, the roc\_auc\_score() function from the sci-kit-learn library is used to compute the AUC of the ROC curve. This function takes the predicted target probabilities for the testing set (y\_pred) and the actual target values for the testing set (y\_test) as input.

The resulting AUC score of 0.9917 indicates that the CatBoost classifier performs exceptionally well in distinguishing crashes involving 2 or more vehicles from those that do not. The high AUC score, approaching the maximum value of 1, suggests that the model is accurate and reliable in its predictions. Thus, this classifier can be considered a highly effective model for predicting whether a crash involves 2 or more vehicles.

**Analysis of CatBoost**

The feature importance graph offers valuable insights into the significant features for predicting whether a crash involves 2 or more vehicles. The importance of each feature is determined by calculating the reduction in the loss function achieved through splitting that particular feature.

A picture containing text, screenshot, diagram, plot

Description automatically generated

Based on the graph, the 'Vehicle\_Configuration' feature emerges as the most crucial factor in predicting the target variable, with a feature importance value of approximately 34%. Following this, in descending order of importance, are 'Crash\_Date,’ 'Crash\_Time,’ 'City\_Town\_Name,’ 'Road\_Surface\_Condition,’ 'Ambient\_Light,’ 'Weather\_Condition,’ 'At\_Roadway,’ and 'At\_Roadway\_Intersect.’

'Vehicle\_Configuration' holds the highest importance value of 34%, while 'At\_Roadway\_Intersect' possesses the lowest value of around 2%. These findings aid in identifying the most pertinent features for predicting the target variable and allow for optimizing the model's performance by focusing on the most significant features.

A screenshot of a computer program

Description automatically generated with medium confidence

The provided code generates a ROC (Receiver Operating Characteristic) curve to assess the binary classification performance of the CatBoost classifier on the crash\_df dataset. The ROC curve visually depicts the relationship between the true positive rate (TPR), or sensitivity or recall, and the false positive rate (FPR) for various classification thresholds. The FPR is calculated as (1-specificity).

A picture containing text, screenshot, line, plot

Description automatically generated

The presented graph illustrates the ROC (Receiver Operating Characteristic) curve of the CatBoost classifier for the testing set, specifically for the class representing crashes involving 2 or more vehicles. The ROC curve is obtained by utilizing the predict\_proba () method of the CatBoost classifier to generate predicted probabilities. The roc\_curve () function from the sci-kit-learn library is employed to calculate the false positive rate (FPR), actual positive rate (TPR), and classification thresholds based on the accurate labels (y\_test) and predicted probabilities (y\_pred).

The matplotlib library plots the ROC curve, with the FPR plotted on the x-axis and the TPR on the y-axis. A gray dashed line represents the performance of a random classifier, and any classifier that outperforms randomness will have a ROC curve positioned above this line.

The graph shows that the CatBoost classifier's ROC curve aligns closely to the top-left corner, indicating its strong performance distinguishing between positive and negative instances. The area under the ROC curve (AUC) is calculated to be 0.9948, meaning outstanding discrimination between positive and negative samples.

In summary, the ROC curve and AUC demonstrate that the CatBoost classifier excels in predicting whether a crash involves two or more vehicles within the crash\_df dataset.

1. What is the trend in the frequency of crashes in Boston between 2012 and 2022?

* The analysis includes a bar chart (Figure 1) showing the number of crashes for every year between 2012 and 2022. This visualization helps identify the trends and patterns in crash occurrences over time.

1. How does the severity of injuries vary among different years?

* The bar chart (Figure 1) categorizes the crashes based on the severity of injuries sustained (fatal or non-fatal), providing insights into the relative risks associated with various types of accidents in different years.

1. How well do the linear and logistic regression models predict the number of vehicles involved in car crashes?

* The analysis mentions performing linear and logistic regression analyses to predict the number of vehicles involved in car crashes. It provides information on the training and testing sets, model instantiation, training, and evaluation. The linear regression model achieved a perfect score of 1.0 on the testing set, indicating accurate predictions based on available variables. The logistic regression model obtained a score of 0.626, suggesting potential for improvement.

1. Did L2 regularization improve the performance of the logistic regression model?

* The analysis mentions applying L2 regularization to the logistic regression model but does not explain whether it improved its performance.

1. How did the logistic regression model perform in binary classification, and what was its accuracy score?

* The analysis mentions transforming the target variable into a binary classification problem and training logistic regression models. The accuracy score for the logistic regression model without regularization was 0.769.

1. How did the linear and logistic regression models perform regarding precision, recall, and F1-score?

* The analysis provides the F1 scores for the linear and logistic regression models. The F1-score for the linear regression model was 0.868, while the logistic regression model achieved a slightly higher score of 0.869.

1. How well did the linear and logistic regression models perform regarding the ROC curve and AUC?

* The analysis provides the ROC curve and AUC values for the linear and logistic regression models. The AUC for the linear regression model was 0.681, indicating better performance compared to the logistic regression model with an AUC of 0.529.

1. How well did the logistic regression model perform regarding the PR-ROC curve and AUC?

* The analysis provides the logistic regression model's PR-ROC curve and AUC value. The AUC for the logistic regression model was 0.776, indicating a favorable balance between precision and recall.

It's important to note that while the analysis mentions specific objectives and analyses, the provided information does not include them.

**Analysis Findings**

The analysis includes information about the performance of linear and logistic regression models in predicting the number of vehicles involved in car crashes. The linear regression model achieved a perfect score of 1.0 on the testing set, indicating accurate predictions. The logistic regression model obtained a score of 0.626, suggesting potential for improvement. However, it does not explicitly mention whether L2 regularization improved the performance of the logistic regression model.

The logistic regression model achieved an accuracy score of 0.769 in binary classification, indicating its effectiveness in classifying whether a crash involves 2 or more vehicles.

The F1-score for the linear regression model was 0.868, while the logistic regression model achieved a slightly higher score of 0.869. These scores indicate the models' performance in terms of precision and recall.

The AUC for the linear regression model was 0.681, indicating better performance compared to the logistic regression model with an AUC of 0.529. The logistic regression model also achieved an AUC of 0.776 for the PR-ROC curve, indicating a favorable balance between precision and recall.

**Regarding the CatBoost analysis:**

The provided code explains the process of using the CatBoost classifier for binary classification on the crash\_df dataset. It covers data splitting, model creation, training, and evaluation using accuracy and AUC scores. This analysis helps answer the question related to the CatBoost classifier's performance in distinguishing crashes involving 2 or more vehicles.

**Interpretations**  
The analysis showed that the linear regression model accurately predicted the number of vehicles involved in car crashes. In contrast, the logistic regression model exhibited the potential for improved precision, and L2 regularization did not significantly improve the logistic regression model's performance. The logistic regression model marginally outperformed the linear regression model regarding the F1 score, but the linear regression model had a higher AUC value in the ROC curve analysis.

These results have important implications for road safety initiatives in Boston. The linear regression model can be valuable for forecasting crashes and informing resource allocation, emergency response planning, and targeted interventions. Although slightly less accurate, the logistic regression model can provide additional insights and precision with further refinement. Leveraging data analysis and predictive modeling can support evidence-based decision-making and the development of proactive road safety measures by considering crash types, temporal patterns, weather impacts, demographics, and geography.

**Recommendations for Further Improving Road Safety Initiatives in Boston:**

1. Enhance the Logistic Regression Model:
   * Refine and fine-tune the logistic regression model for improved predictive performance.
   * Explore feature engineering techniques, regularization parameter adjustments, and alternative classification algorithms.
2. Explore Additional Variables:
   * Incorporate road infrastructure characteristics, driver behavior data, and information on traffic control devices into future analyses.
   * These variables can provide valuable insights into crash causes and aid in developing targeted interventions.
3. Incorporate Temporal and Spatial Analysis:
   * Conduct a more detailed temporal analysis by examining crash patterns by hour or day of the week.
   * Utilize spatial analysis techniques to identify crash hotspots and allocate resources effectively.

Overall, these recommendations aim to improve the accuracy and precision of predictive models, enhance understanding of crash factors, and support evidence-based decision-making in road safety initiatives.

To gain a comprehensive understanding of crash occurrences, it is recommended to incorporate additional variables in future analysis. These variables include road infrastructure characteristics, driver behavior data, traffic control devices, environmental factors, vehicle characteristics, and urban design/land use. Considering these variables, the analysis can provide valuable insights into the causes of crashes and facilitate the development of targeted interventions to improve road safety.

**Conclusion**

In conclusion, analyzing linear regression, logistic regression, and CatBoost models for predicting car crashes in Boston provides valuable insights for road safety initiatives. The linear regression model exhibits exceptional predictive capability, enabling efficient resource allocation and emergency response planning. The logistic regression model shows promise in improving precision for crash occurrence predictions. The CatBoost model accurately distinguishes between crashes involving 2 or more vehicles and those involving fewer vehicles. Combining the insights from all three models can further enhance road safety measures.

Evaluation metrics indicate that the linear regression model possesses superior discrimination power, while the logistic regression model achieves a slightly better F1 score. Integrating additional variables such as road infrastructure characteristics, driver behavior, and environmental factors can lead to a comprehensive understanding of crash occurrences. It is recommended to continuously refine the models and collaborate with stakeholders to implement targeted interventions effectively, ultimately reducing car crashes in Boston and enhancing road safety.

**References:**

*Vision zero crash records*. Analyze Boston. (n.d.). https://data.boston.gov/dataset/vision-zero-crash-records

Hastie, T., Tibshirani, R., & Friedman, J. (n.d.). *The Elements of Statistical Learning*. SpringerLink. <https://link.springer.com/book/10.1007/978-0-387-84858-7>

Google. (n.d.-a). *Applied Logistic Regression*. Google Books. <https://books.google.com/books?hl=en&lr=&id=64JYAwAAQBAJ&oi=fnd&pg=PR13&dq=Hosmer%2C%2BD.%2BW.%2C%2BLemeshow%2C%2BS.%2C%2Band%2BSturdivant%2C%2BR.%2BX.%2B%282013%29.%2BApplied%2BLogistic%2BRegression%2B%283rd%2Bed.%29&ots=DtjO6W5qlL&sig=K6XZyt7GSZlj24ubmpOxipiOBTU#v=onepage&q=Hosmer%2C%20D.%20W.%2C%20Lemeshow%2C%20S.%2C%20and%20Sturdivant%2C%20R.%20X.%20(2013).%20Applied%20Logistic%20Reghttps://books.google.com/books?hl=en&lr=&id=64JYAwAAQBAJ&oi=fnd&pg=PR13&dq=Hosmer,+D.+W.,+Lemeshow,+S.,+and+Sturdivant,+R.+X.+(2013).+Applied+Logistic+Regression+(3rd+ed.)&ots=DtjO6W5qlL&sig=K6XZyt7GSZlj24ubmpOxipiOBTU#v=onepage&q=Hosmer%2C%20D.%20W.%2C%20Lemeshow%2C%20S.%2C%20and%20Sturdivant%2C%20R.%20X.%20(2013).%20Applied%20Logistic%20Regression%20(3rd%20ed.)&f=falseression%20(3rd%20ed.)&f=false>

Washington, T. C. U. of, Chen, T., Washington, U. of, Washington, C. G. U. of, Guestrin, C., IBM, Bosch, Amazon, Baidu, & Metrics, O. M. A. (2016, August 1). *XGBoost: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and data mining*. ACM Conferences. <https://dl.acm.org/doi/10.1145/2939672.2939785>

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (1970, January 1). *CatBoost: Unbiased boosting with categorical features*. Advances in Neural Information Processing Systems. <http://papers.neurips.cc/paper/7898-catboost-unbiased-boosting-with-categorical-features>