Hybrid Approach for Crash Severity Prediction using XAI Methods

Prudhvi Sai Keerthi
Department of Data Science
University of North Texas
Denton, Texas
prudhvisaikeerthi@my.unt.edu

Dheeraj Suryadevara
Department of Data Science
University of North Texas
Denton, Texas
dheerajsuryadevara@my.unt.edu

Sreetej Asuri Maringanti
Department of Data Science
University of North Texas
Denton, Texas
sreetejasurimaringanti@my.unt.edu

Vinay Krishna Gudla
Department of Data Science
University of North Texas
Denton, Texas
vinaykrishna@my.unt.edu

***Abstract*—This study introduces an explainable artificial intelligence (XAI) approach-based hybrid approach for crash severity prediction. Random Forest exhibited the best performance, followed by logistic regression, and support vector machines were trained for data preprocessing, modeling, and hyperparameter tuning. Among them, vehicle make and type were key features identified by Permutation Importance. Metrics such as diversity, proximity, and allowed ranges were used to generate counterfactual explanations, giving personalized insights. Due to these shape/techniques differences, Random Forest distance metrics were smallest as well, pointing out that those techniques provide actionable counterfactuals. This approach employs predictive modeling and XAI to improve model accuracy and interpretability on severity prediction of crashes.**

***Index Terms*—Counterfactuals, Permutation Importance, Crash severity, Classification.**

I. Introduction

Crash severity prediction is important for factors like improving road safety, reducing injuries, and others. Building a prediction model to accurately predict the outcomes can help the authorities and emergency responders to identify the reasons for the increase in crash severity and provide opportunities to implement preventive measures to avoid the crashes. Moreover, the awareness given based on the data driven insights helps in educating drivers on high-risk behaviors and conditions. This project builds a prediction model using machine learning, and explainable AI techniques helps to pinpoint the factors that influence the crash severity outcomes. By combining two methods permutation importance to select important features and counterfactual explanations to get the explanations on the outcomes obtained in the prediction. By generating counterfactual explanations automotive experts can identify the features that need to be modified for reducing the crash severity.

A. Explainable AI Techniques XAI

XAI is a set of procedures and methods that permit the human users to understand and trust the results and outputs obtained by machine learning models/algorithms. XAI is something that describes an AI model, its expected impact and potential biases. This helps to distinguish the features like model accuracy, fairness, transparency and outcomes in AI-powered decision making [1].

XAI also helps an organization to select a suitable approach to AI development because it is important for an organization to understand how an AI decision making process works and accountability of AI so as not to trust the AI model decisions blindly. XAI can guide humans to understand and explain machine learning algorithms, deep learning algorithms, and neural networks. Machine learning models are considered as black boxes that are impossible to understand or interpret. Similarly neural networks used in deep learning are said to be some of the hardest for a human to understand.

AI models' results and performance may drift from the original because the training data differs from production data [2]. It ultimately makes it important for the organizations to continuously monitor and manage models to promote XAI while measuring its performance and impact with business with such algorithms. It helps to mitigate legal, compliance, security and improve the end users trust towards the use of AI.

The explainability techniques are primarily divided into two categories global and local. Global explanations explains the model's generic operating rules [4]. Whereas local explanations explains for every single data how the model had obtained a certain output and what are the rules that supported it.

Overall, this project demonstrates the significance of leveraging XAI methods with predictive modeling for crash severity prediction. The work provides the actionable insights that can support automotive designers and safety analysts in making data-driven decisions to minimize crash severity.

II. Review of Literature

Lorenzo Tronchin et al. (2021) proposed a paper highlighting the importance of Explainable Artificial Intelligence methods in model interpretability. The paper covers techniques such as shapelets, prototypes, LIME, Grad-CAM, and Integrated Gradients. Authors faced some hurdles in adapting these methods from images to MTS. The study covers limited research on MTS-specific XAI methods, and new datasets and techniques are needed to understand spatial and complex temporal dependencies [9].

The research paper by Michael Kamp et al. (2021) encompasses the developments in XAI for time-series forecasting using TS-MULE, which is an extension of LIME. This approach utilizes the latest segmentation techniques specific to time series to increase the interpretability of deep learning models. The accuracy of local explanations is improved with this approach, and the problems associated with multivariate data in different domains are addressed [8].

Amit Sharma et al. (2020) introduced a framework for generating diverse counterfactual explanations for different machine learning classifiers. They tuned feasibility and diversity with the help of a detrimental point process for developing better counterfactuals. Researchers used real-world datasets for evaluation, and local decision boundary approximation was improved compared to methods like LIME [7].

Giovanni et al. (2024) surveyed Graph Counterfactual Explanations (GCE) and provided formal definitions, benchmarks, and evaluation methods. Fourteen methods were reviewed on 22 datasets with the help of 19 metrics. GRETEL library was also used to perform empirical analysis. They also highlighted the problems with fairness, minimality, and privacy in generating counterfactual explanations for graph neural networks [6].

Matan Atad et al. (2024) used Diffusion Autoencoders (DAEs) to generate the counterfactual explanation in the medical imaging domain. Unsupervised feature extraction in latent space was used to facilitate regression and classification tasks, including diabetic retinopathy and vertebral fractures. This approach improves interpretability by suppressing the problems related to model bias and label scarcity in clinical applications [5].

III. Objectives of the Study

- To classify crash severity using Blackbox model by leveraging XAI methods for interpretability.
- To find the important features that impact crash severity using Permutation Importance and Counterfactual Explanations.
- To provide the small changes using Counterfactual Explanations for reducing the crash severity.
- To enhance the explainability of crash prediction models for good decision-making.

IV. Methodology

First step in the project is to load and prepared the data using the pandas, scikitlearn, SMOTE packages. Then EDA is performed using seaborn and matplotlib packages. After preparing data, data is splitted to training and testing for model traing. In this step, three classifier model Random Forest Classifier, Logistic Regression, and Support Vector Machines are used. Later, Permutation Importance is used for identifying the important features. Eventually, counterfactuals are generated using DICE and evaluated using distance metrices, Recourse, and losses.

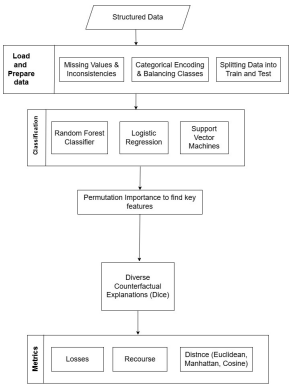

Fig. 1. Flowchart to generate Counterfactuals

V. Data Collection

Synthetic Indian Automobile Crash Detection dataset is collected from Kaggle. The dataset is used to analyze and predict the automobile crashes in India. The dataset has 25 attributes and has 10,000 which makes it good for building predictive models. The dataset has key features like driver demographics, vehicle details, environmental conditions, road characteristics and crash severity. Dataset Link: https://www.kaggle.com/datasets/swish9/synthetic-indian-automobile-crash-data

VI. Data Cleaning

A. Handling Missing Values and duplicates

Handling missing values is crucial in data analysis and machine learning modeling because it affects the model predictions' reliability, accuracy, and interpretability. In some cases, inadequate cleaning tends to biased results. Missing values in the dataset mean it does not represent the actual population. To check the duplicates, we used df.duplicated().sum() and found 0 duplicates in the dataset.

We used the df. isnull().sum()*100/len(df) to calculate the percentage of the missing value in each column. It was found that vehicle make, vehicle type, engine type, and some other columns have zero missing values. Columns like ABS Presence and ESC Presence have missing values of around 7-10%. The highest percentage of missing values is found in Day of week (12.71%).

We handled the missing values in the dataset by filling the mode because all the missing values are present in categorical columns. Also, we filled the null values of the gender column with the value "Others." By doing this, we can ensure consistency and avoid data loss, thus providing a complete dataset for the analysis.

B. Handling Inconsistencies

There were some inconsistencies in the binary presence columns (ABS presence, TPMS presence, ESC presence, TCS presence). So, all " True " values were replaced with numeric value 1. This provides a better representation of data for machine learning algorithms and data analysis.

VII. Exploratory Data Analysis

A. Correlation Analysis on Numerical Data

Correlation Analysis is a statistical technique for measuring the dataset's relationship between two numerical features. By doing this, we can find the variable that affects other variables with more minor changes. The correlation coefficient ranges from -1 to 1, and 1 indicates a strong relationship between variables. On the other hand, -1 means negative, and zero means no correlation between variables. From the figure below, we can observe that most attributes have a weak correlation with others, and there was a moderate correlation between ABS presence, ESC presence, TPMS presence, and TCS presence.

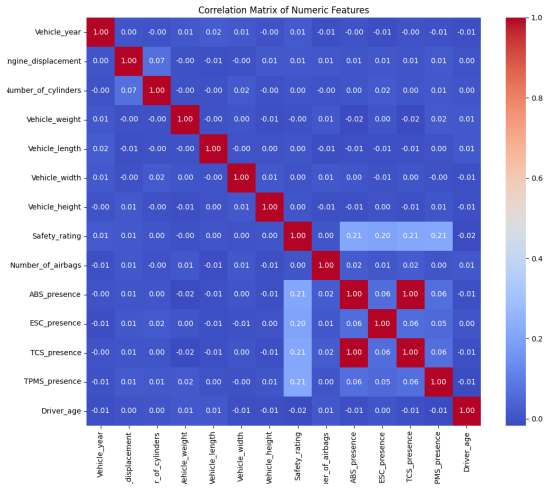

Fig. 2. Correlation analysis on numerical data

B. Correlation Analysis on Categorical Data

After performing correlational analysis on numerical data, we performed correlational analysis on categorical data using Cramer's V correlation analysis. This analysis is used to measure the strength of association between categorical attributes. This method is employed because Pearson's correlation is not applicable for categorical variables. This method creates a contingency table for two categorical variables, and then the chi-squared statistic is calculated. The values range between 0 (no association) and 1 (strong association). Later, it was found that vehicle make and Crash severity had a high correlation with a value of 0.67. Meanwhile, Transmission type and weather conditions have a correlation nearer to zero, which represents independence.

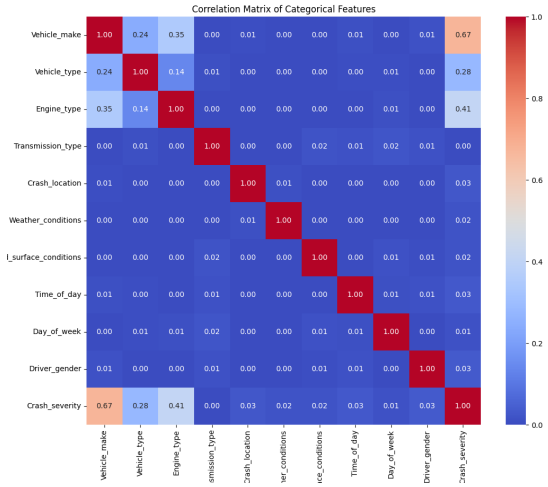

Fig. 3. Correlation analysis on categorical data

C. Outlier Analysis

Outlier analysis is performed using the IQR method to identify the data points deviating from the other observations. It was found that most of the numerical columns do not have any outliers representing consistent data. However, the binary columns show higher outlier counts because of their nature, which means they are not outliers.

D. Univariate Analysis

Univariate analysis is conducted on the dataset to understand the distribution and frequency of the columns. The analysis found that more crashes occurred on dry roads and under clear weather conditions. Also, crashes were frequent on Wednesdays and during afternoons and evenings. Sedans and hatchbacks are two vehicle types involved in more crashes than other vehicle types. Other variables, like the number of airbags and safety ratings, have balanced distributions. The number of cylinders with 6 and 4 are predominant in crashes.

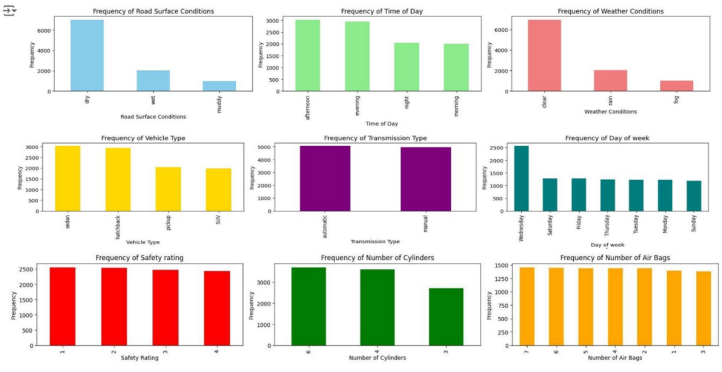

Fig. 4. Univariate Analysis on Categorical Data

The figure below illustrates the histograms of important features such as Engine displacement, vehicle weight, vehicle

length, driver age, and vehicle width. It was found that many vehicles in the dataset have average engine size, moderate length and weight. Most of the drivers involved in the accidents are younger to middle-aged.

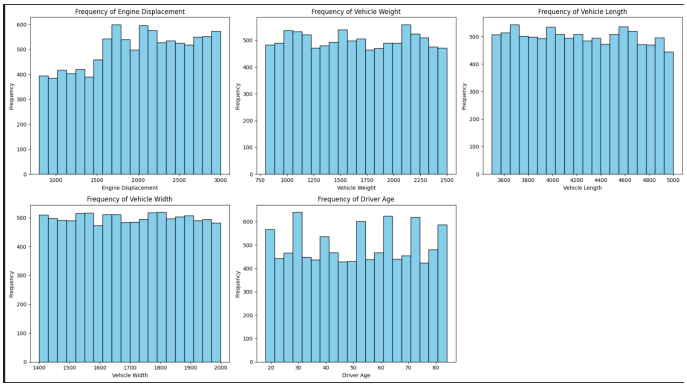

Fig. 5. Univariate Analysis on Numerical Data

E. Bivariate Analysis

We also performed bivariate analysis using box plots to explore the relationship between important vehicle features and target variable crash severity.

After that, a figure was plotted to determine the target variable crash severity distribution. A primary number of records is related to the severe category. Minor contributes only 20 records, so those are merged with moderate because of significantly less count.

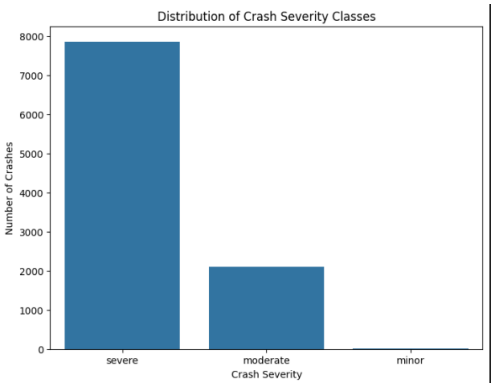

Fig. 6. Distribution of Target Variable

VIII. Statistical Testing

- Null hypothesis is there is no significant relationship between our target variable and other variables in the dataset.
- Alternative hypothesis is there is relationship between target variable and other variables

For that we performed Chi-Square test and ANOVA test (Analysis of Variance). From both the tests it was found that vehicle make, vehicle type, engine type, vehicle year, engine displacement, and number of cylinders are having significant association with the target variable.

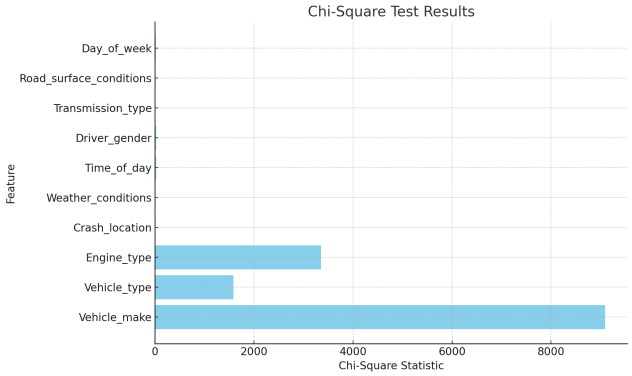

Fig. 7. Chi square test results

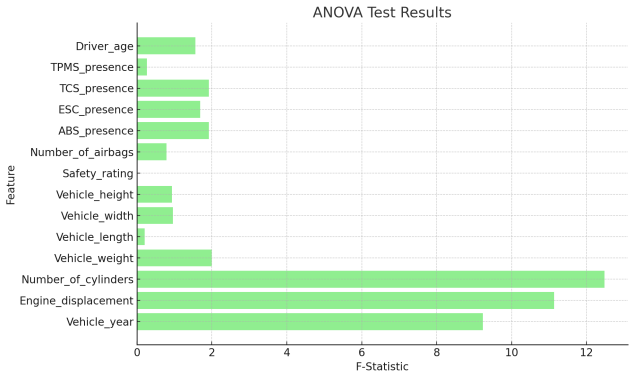

Fig. 8. ANOVA test results

IX. SMOTE for Handling Class Imbalance

Synthetic Minority Over-sampling Technique) is an oversampling technique used to solve class imbalance problems, precisely classification problems. This technique generates synthetic samples for the minority class instead of duplicating the existing records. It uses minority class instances and creates new samples by interpolating between the minority and its nearest neighbors. SMOTE helps us prevent model bias towards the majority class and avoids overfitting the model. The figure below shows the frequency of each class after using the SMOTE technique.

X. Converting Categorical to Numerical Variables

The conversion of categorical variables to numerical ones is significant because of model compatibility, mathematical operations, and performance issues. Numerical data will allow models to use mathematical operations for model training and optimization. In this project, a Label encoder was employed to encode variables 'Vehicle make,' 'Vehicle type,' 'Driver gender,' and 'Engine type.' Then, we used mapping for some ordinal categorical variables, as shown below.

Fig. 9. Univariate Analysis on Numerical Data

XI. Model Training

A. Splitting Data

Fo training the model, data is splitted to 80% training data and 20% testing data. We used train_test_split function from scikit learn to split the data.

B. Random Forest Classifier

This algorithm is used quite a lot in machine learning today. The outputs of various decision trees are combined to render only one result. It is also called the ensemble method Random Forest model and is applicable for classification as well as regression problems.

The random forest method is the advanced version of the bagging method since it combines bagging and randomness, that is why it is also called "the random subspace method" or feature bagging. Random subsets of input features are generated making sure that there is low correlation between the subsets (decision trees).

In random forest, we tuned hyperparameters, which basically stand at three key hyperparameters, namely those with respect to node size, number of trees, and number of features. This tuning helps in the way of selecting a model with the best combination of hyperparameters for its best performance on the accuracy side in terms of what is to be predicted.

The values we opted in the hyperparameter tuning are:
'n_estimators': [50, 100, 200],
'max_depth': [None, 10, 20],
'min_samples_split': [2, 5, 10],
'min_samples_leaf': [1, 2, 4]

C. Logistic Regression

A statistical model, also called logit model, it is used for classification and prediction analytics. The dependent variable always resulted between 0 and 1. In this model, a logic is used for logit transformation on the odds, i.e., the probability of success divided by the probability of failure.

D. Support Vector Machines

SVMs are one of the supervised machine learning algorithms. It allows the classification of the given data through the optimal line or hyperplane that maximizes the distance from one another for each class in an N-dimensional space. Usually, SVMs are employed to solve classification problems. It classifies between two classes searching for optimally determined hyperplane maximizes margin on two opposite classes' closest data points. The input's feature count will either determine if the hyperplane is a line in 2-D space or a plane in an n-dimensional space. The fact that many hyperplanes are possible teaches us that maximizing the margin leads to the definition of the most appropriate decision boundary across the classes. This, in effect, leads to a generalization concerning the new data with high accuracy and prediction classifications. The lines adjacent to the optimal hyperplane are called support vectors because these vectors run through the data points that determine maximum margin.

E. Permutation Importance

Permutation feature importance is a model-agnostic technique that quantifies the importance of a given feature to the performance of the overall model. The explanation will involve randomly shuffling the values of one feature at a time and monitoring the deterioration in the model's score as a means to reveal how much that model relies on the respective feature. This method can be applied with any fitted model, which, in turn, makes this an extremely generic method across many model types, including nonlinear and inherently complex models. Running this process, several times returns an idea of the method variance in the feature importances, too. Let's not forget to mention here that one should first check the predictive performance of their model on a holdout set or with cross-validation, since poor performance models may return untrustworthy feature importances. The idea of permutation feature importance cannot be interpreted as a given feature's inherent predictive power, but it is a measure of the feature's importance about the model that is used. The permutation_importance function calculates importance using feature shuffling and multiple trials.

F. Counterfactual Explanations

Counterfactual explanations are defined as describing causality in terms of 'what if', such that, "If X had not happened, Y would not have happened." Example: "If I had not drunk this hot coffee, I would not have burned my tongue." Herein, X is the cause, and Y is the event. Thus, counterfactual explanations in interpretable machine learning clearly define predictions for single instances. Here, the "event" is the predicted outcome, while the "causes" are the input feature values that determine this prediction. These explanations identify which feature changes drive different outcomes and enhance model interpretability. They can serve for actionable insights, fairness, and to make sure models produce transparent, understandable, and accountable decisions. We used DICE package in our project to generate the counterfactuals.

1) Diversity: By tuning the diversity, variety of counterfactuals are generated. Using this parameter, range of possible alternatives can be found. In this study we use 1 value for diversity.

2) Proximity: refers to how similar the generated alternative is with respect to query instance. By customizing this parameter, realistic and feasible counterfactuals can be generated and by staying within reasonable boundaries. In our study, we opted proximity as 1.

3) Permitted Range: This parameter refers to the constraints for the features of a counterfactual instance. By using these ranges, we can able to maintain the domain relevance and feasibility.

```
permitted_range = {
    'Driver_age': [12, 100],
    'Number_of_airbags': [0, 10],
    'Vehicle_weight': [700, 3000],
    'Vehicle_length': [3000, 6000],
    'Vehicle_width': [1500,2500],
    'Engine_displacement': [800, 8000],
    'Number_of_cylinders':[3,16],
    'Safety_rating': [1, 5]
}
```

Fig. 10. Permitted Range

4) Features to vary: In the context of diverse counterfactuals, it refers to the attributes we can alter to achieve the desired outcome. We used below features to alter the target outcome.

```
features_to_vary = ['Engine_displacement', 'Vehicle_weight', 'Vehicle_length', 'Vehicle_width',
                    'Number_of_airbags', 'Number_of_cylinders', 'Safety_rating',
                    'Transmission_type', 'Crash_location',
                    'Road_surface_conditions',
                    'Engine_type']
```

Fig. 11. Features to vary

For evaluating the counterfactuals we used metrics like distances, losses, and recourse. Eucliden distance calculates the straight-line distance between two points whereas, manhattan distance calculates the distance by adding the absoluted differences of their coordinates. Cosine distance uses dissimilarity between two vectors by determining the cosine of angle between them.

Recourse or Change score measures how many feature need to change to alter the target variable. Delta values are the difference between original value and counterfactual value [3].

XII. Results

A. Random Forest Classifier

The best combination resulted is 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100.

From the below images it was evident that random forest model performed well with higher accuracy and AUC of 1 which indicates optimal classification. The confusion matrix of random forest model is also well balanced indicating good predictive performance.

```
Best hyperparameters: {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}
Accuracy: 0.9822109275730623
              precision    recall  f1-score   support

           0       1.00      0.97      0.98      1509
           1       0.97      1.00      0.98      1639

    accuracy                           0.98      3148
   macro avg       0.98      0.98      0.98      3148
weighted avg       0.98      0.98      0.98      3148
```

Fig. 12. RandomForest Classifier Results.

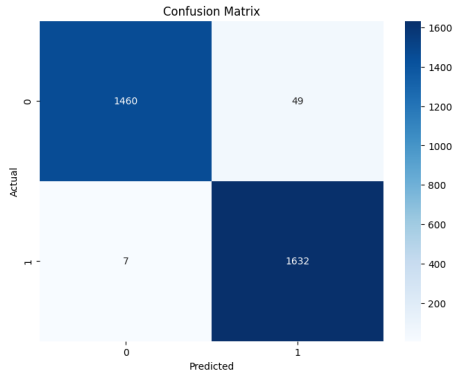

Fig. 13. Confusion matrix of RandomForest Classifier.

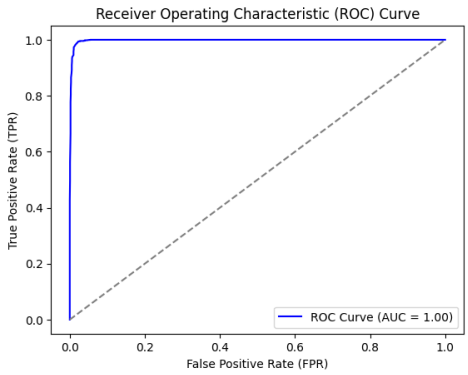

Fig. 14. ROC curve of RandomForest Classifier.

B. Logistic Regression

The ROC curve of Logistice Regression is ideal representing good performance. AUC value of 0.90 means, the model has high probability of classifying records correctly. The confusion matrix of logistic regression was also well balanced between true positives and true negatives.

```
Logistic Regression Accuracy: 0.8459339263024143
              precision    recall  f1-score   support

           0       0.80      0.90      0.85      1509
           1       0.90      0.79      0.84      1639

    accuracy                           0.85      3148
   macro avg       0.85      0.85      0.85      3148
weighted avg       0.85      0.85      0.85      3148
```

Fig. 15. Logistic Regression Results

C. Support Vector Machine model

For SVM also the ROC curve is ideal with AUC of 0.90. Whereas, the model predicted more False positives which is the limitation for this model. Overall, it has high trye positive rate compared to false positive rate.

Overall, the Random Forest classifier got good results in terms of accuracy and other metrics compared to other metrics.

D. Permutation Importance

After model training each model is passed through Permutation Importance method to know the importance features

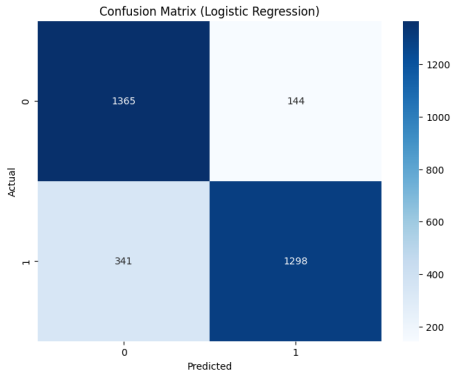

Fig. 16. Confusion matrix of Logistic Regression Model.

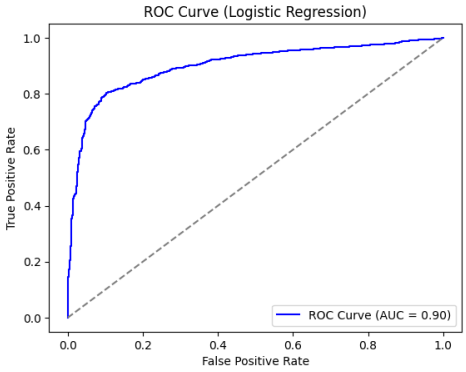

Fig. 17. ROC off Logistic Regression Model.

SVM Accuracy: 0.7175984752223634

	precision	recall	f1-score	support
0	0.89	0.47	0.61	1509
1	0.66	0.95	0.78	1639
accuracy			0.72	3148
macro avg	0.78	0.71	0.70	3148
weighted avg	0.77	0.72	0.70	3148

Fig. 18. SVM Model Results

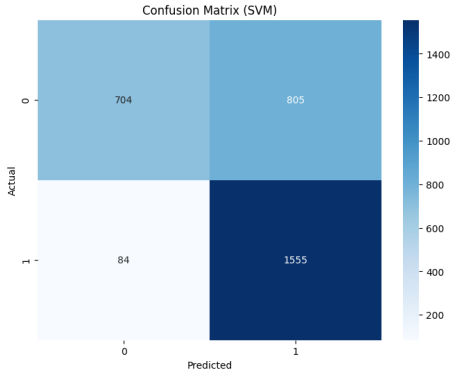

Fig. 19. Confusion matrix of SVM Model.

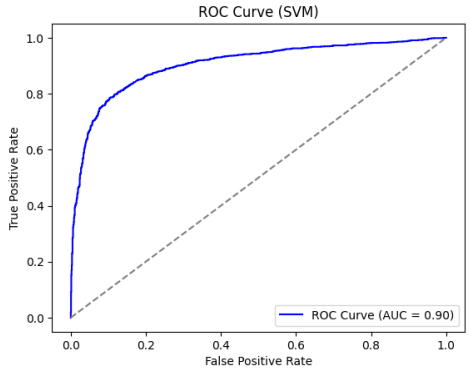

Fig. 20. ROC of SVM Model.

for each model. The below graph represents the permutation importance for three models. It was found that Vehicle Make and Vehicle Type played key role in predicting crash severity for random forest and logistic regression model. Whereas, SVM predicted target variable using many influential features and represents the sensitivity to multiple inputs.

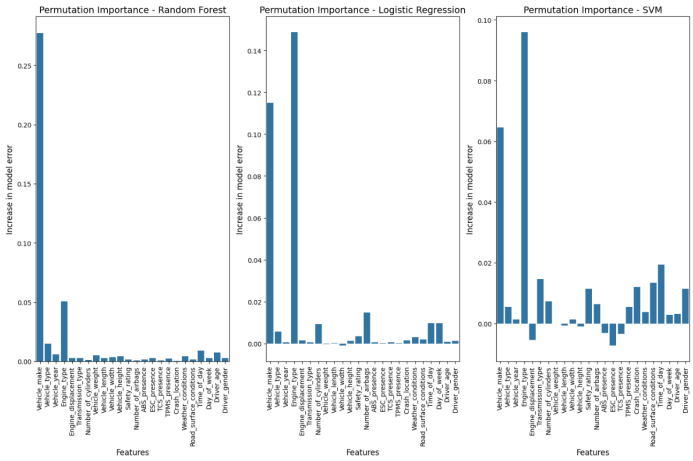

Fig. 21. Confusion matrix of SVM Model.

E. Counterfactual Explanations

From permutation importance significant features for each model was found and now counterfactuals are generated for calculating the local importance. Counterfactuals are generated for each model by considering one query instance from the test data. Then for all the models distance metrics, Change score (Cscore), losses are calcuated. In the below figures we can see the metrics related to the counterfactuals.

In the below figure it was found that counterfactuals generated for random forest classifier has less distance between the original and generated counterfactual compared others.

Then losses are calculated by summation of proximity loss and sparsity loss. For this also Random forest got a loss of 332 which is lower compared to SVM (3606) and Logistic Regression (1450).

```
Transposed Query Instance:
                              10870
Vehicle_make                2.000000
Vehicle_type                0.000000
Vehicle_year             2021.000000
Engine_type                 2.000000
Engine_displacement      1722.000000
Transmission_type           0.000000
Number_of_cylinders         3.000000
Vehicle_weight           2320.000000
Vehicle_length           4124.000000
Vehicle_width            1620.000000
Vehicle_height           1731.000000
Safety_rating               3.000000
Number_of_airbags           5.000000
ABS_presence                1.000000
ESC_presence                0.783502
TCS_presence                1.000000
TPMS_presence               1.000000
Crash_location              1.000000
Weather_conditions          0.000000
Road_surface_conditions     0.000000
Time_of_day                 0.000000
Day_of_week                 1.000000
Driver_age                 67.000000
Driver_gender               0.000000
```

Fig. 22. Query Instance used

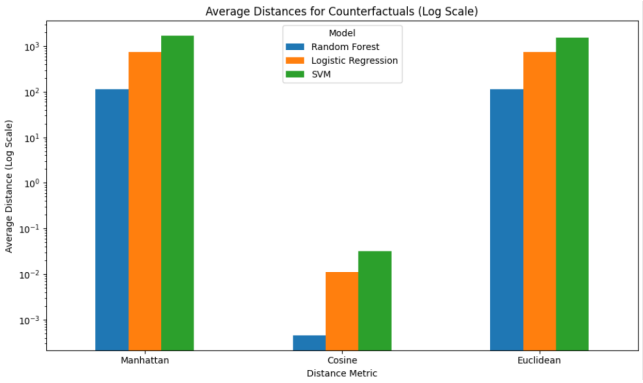

Fig. 23. Average distances for counterfactuals.

```
Random Forest Losses:
Proximity Loss: 326.0245539689523
Sparsity Loss: 6
Total Loss: 332.0245539689523

Logistic Regression Losses:
Proximity Loss: 1444.5061518822667
Sparsity Loss: 6
Total Loss: 1450.5061518822667

SVM Losses:
Proximity Loss: 3599.726817777003
Sparsity Loss: 7
Total Loss: 3606.726817777003
```

Fig. 24. losses for counterfactuals.

For all three models CScore was same (2) which means only two features are altered for changing the target variable to opposite label.

```
Counterfactual 0 - CScore: 2, Delta Values: {'Number_of_cylinders': 6.0, 'Vehicle_weight': 1150.0}
Counterfactual 1 - CScore: 2, Delta Values: {'Number_of_cylinders': 12.0, 'Number_of_airbags': 2.0}
Counterfactual 2 - CScore: 2, Delta Values: {'Engine_type': 2.0, 'Safety_rating': 1.0}

Counterfactuals Using Random Forest Classifier

Counterfactual 0 - CScore: 2, Delta Values: {'Engine_type': 1.0, 'Vehicle_length': 694.0}
Counterfactual 1 - CScore: 2, Delta Values: {'Vehicle_weight': 1439.0, 'Crash_location': 1.0}
Counterfactual 2 - CScore: 2, Delta Values: {'Vehicle_weight': 126.0, 'Safety_rating': 2.0}

Counterfactuals Using Logistic Regression

Counterfactual 0 - CScore: 2, Delta Values: {'Engine_displacement': 1228.0, 'Vehicle_width': 310.0}
Counterfactual 1 - CScore: 2, Delta Values: {'Vehicle_width': 489.0, 'Road_surface_conditions': 2.0}
Counterfactual 2 - CScore: 2, Delta Values: {'Vehicle_weight': 537.0, 'Vehicle_width': 810.0}

Counterfactuals Using Support Vector Machines
```

Fig. 25. Cscore and Delta values

XIII. Conclusion

Overall, the performance and evaluation results signify that the random forest model shows the maximum accuracy for predicting the crash severity, and high scores for F1, Recall, Precision, with lowest log loss value. With further analysis using explainable AI techniques like permutation importance the important features that are contributing to the model performance are identified with the amount of model error generated with shuffling of single feature values in multiple combinations for all the features. With the help of counterfactual explanations, we can see various results for each prediction that can help us to identify or learn what features can be modified within the permissible limits to attain a desired prediction, like if the originally predicted severity is minor, what features can be modified to make a prediction that is severed. Learning this change in feature values will help automobile companies, designers, or automobile engineers build a safe vehicle to avoid severe crash severity.

Also, Random Forest classifier performed well in generating counterfactuals with less distance, CScore, and losses. However, this prediction alone and using the explanations doesn't justify that the results generated can be used to build a vehicle without errors.

References

[1] Gunonu, S., Altun, G., & Cavus, M. (2024). Explainable bank failure prediction models: Counterfactual explanations to reduce the failure risk. arXiv preprint arXiv:2407.11089.

[2] Li, J., Yang, Y., Hu, Y., Zhu, X., Ma, N., & Yuan, X. (2023). Using multidimensional data to analyze freeway real-time trac crash precursors based on xgboost-shap algorithm. Journal of Advanced Transportation, 2023(1), 5789573.

[3] Nirmalraj, S., Antony, A. S. M., Srideviponmalar, P., Oliver, A. S., Velmurugan, K. J., Elanangai, V., & Nagarajan, G. (2023). Permutation feature importance-based fusion techniques for diabetes prediction. Soft Computing, 112.

[4] Guidotti, R. (2024). Counterfactual explanations and how to find them: literature review and benchmarking. Data Mining and Knowledge Discovery, 38(5), 2770-2824.

[5] Atad, M., Schinz, D., Moeller, H., Graf, R., Wiestler, B., Rueckert, D., ... & Keicher, M. (2024). Counterfactual Explanations for Medical Image Classification and Regression using Diffusion Autoencoder. arXiv preprint arXiv:2408.01571.

[6] Prado-Romero, M. A., Prenkaj, B., Stilo, G., & Giannotti, F. (2024). A survey on graph counterfactual explanations: definitions, methods, evaluation, and research challenges. ACM Computing Surveys, 56(7), 1-37.

[7] Mothilal, R. K., Sharma, A., & Tan, C. (2020, January). Explaining machine learning classifiers through diverse counterfactual explanations. In Proceedings of the 2020 conference on fairness, accountability, and transparency (pp. 607-617).
[8] Kamp, M., Koprinska, I., Adrien, B., Bouadi, T., Frénay, B., Galárraga, L., ... & Lijffijt, J. (2021). Machine learning and principles and practice of knowledge discovery in databases.
[9] Tronchin, L., Sicilia, R., Cordelli, E., Celsi, L. R., Maccagnola, D., Natale, M., & Soda, P. (2021, October). Explainable ai for car crash detection using multivariate time series. In 2021 IEEE 20th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC) (pp. 30-38). IEEE.