Accident Severity Prediction

Precented By: Anas Pathan

Content

- Abstract
- Introduction
- Approach
- Data cleaning
- EDA
- Experimental section
- Model Selection
- Conclusions

Abstract

- The goal of this project is to build a machine learning model capable of accurately predicting the severity of car accidents. By leveraging real-world data, they aim to provide valuable insights to emergency responders and enable proactive measures for mitigating accident severity.
- Road accidents are a major public safety concern, causing thousands of fatalities and injuries every year. Accurately predicting accident severity can help emergency responders allocate resources more efficiently and assist policymakers in improving road safety. This project aims to build a machine learning model that predicts the severity of accidents (fatal vs. non-fatal) using real-world collision data.
- Multiple machine learning models, including Logistic Regression and Random Forest, were implemented and evaluated using performance metrics such as accuracy, precision-recall, F1-score, and AUC-ROC curves

Introduction

- Traffic accidents are a major public safety concern, leading to injuries, fatalities, and significant economic losses. Predicting accident severity can help emergency responders prioritize resources, improve road safety policies, and reduce casualties.

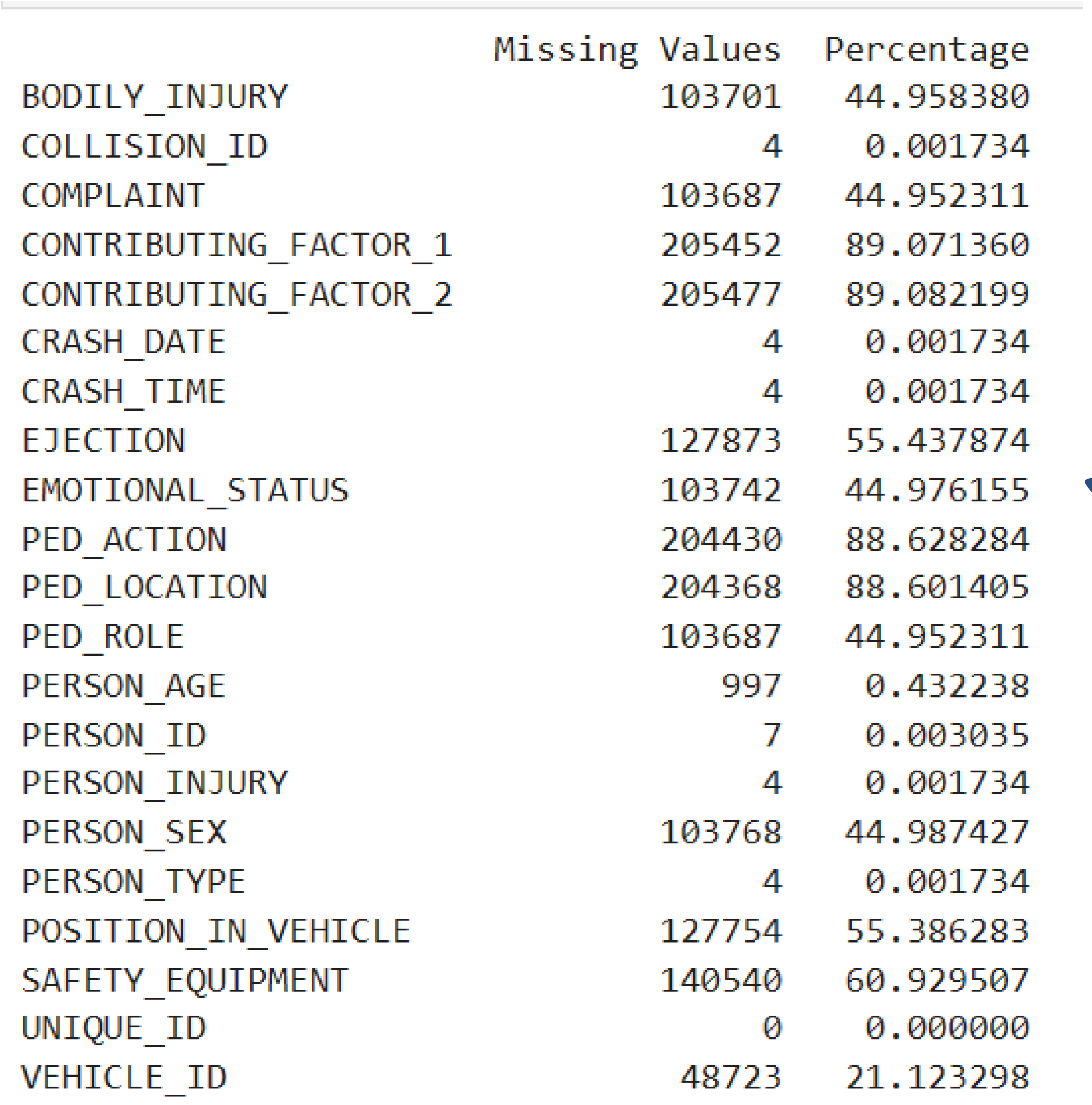
- This project leverages machine learning to analyze accident data and predict injury severity based on various factors, including crash time, person type, emotional status, safety equipment usage, and more. By applying data preprocessing, feature selection, and classification models, we aim to develop an accurate predictive model that can assist in proactive decision-making for road safety improvements.

Approach - We are following below approach to create our accident prediction model which is capable of predicting the number of casualties as per business requirements.

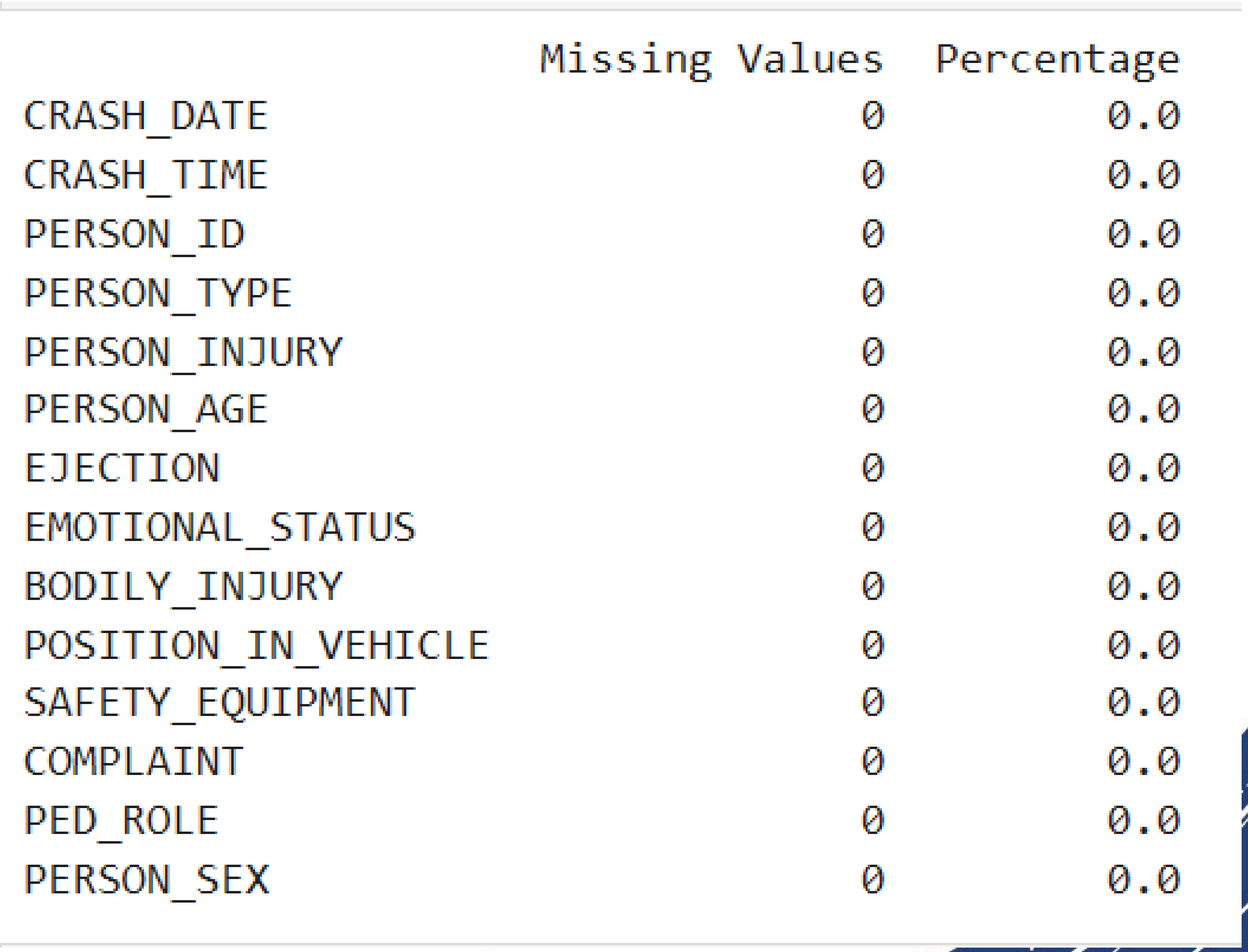
1. **Data Collection & Cleaning – Gathering real-world accident data, handling missing values, and preprocessing categorical variables.**
2. **Exploratory Data Analysis (EDA) – Identifying key patterns, visualizing accident trends, and understanding feature relationships.**
3. **Feature Engineering – Transforming raw data into meaningful features, encoding categorical variables, and scaling numerical data.**
4. **Model Selection & Training – Implementing and evaluating different machine learning models, including Logistic Regression, Random Forest, and XGBoost.**
5. **Performance Evaluation – Assessing model effectiveness using accuracy, precision, recall, F1-score, and AUC-ROC metrics.**
6. **Insights & Optimization – Analyzing key influencing factors, improving model performance, and making data-driven recommendations.**

Cleaning Part

- First we have to Clean the data becouse the data that we working have a large amount of Nan values, Have missing Values, Some imbalance data So first need to clean that.
- Here is the Demo of Before and After of data cleaning



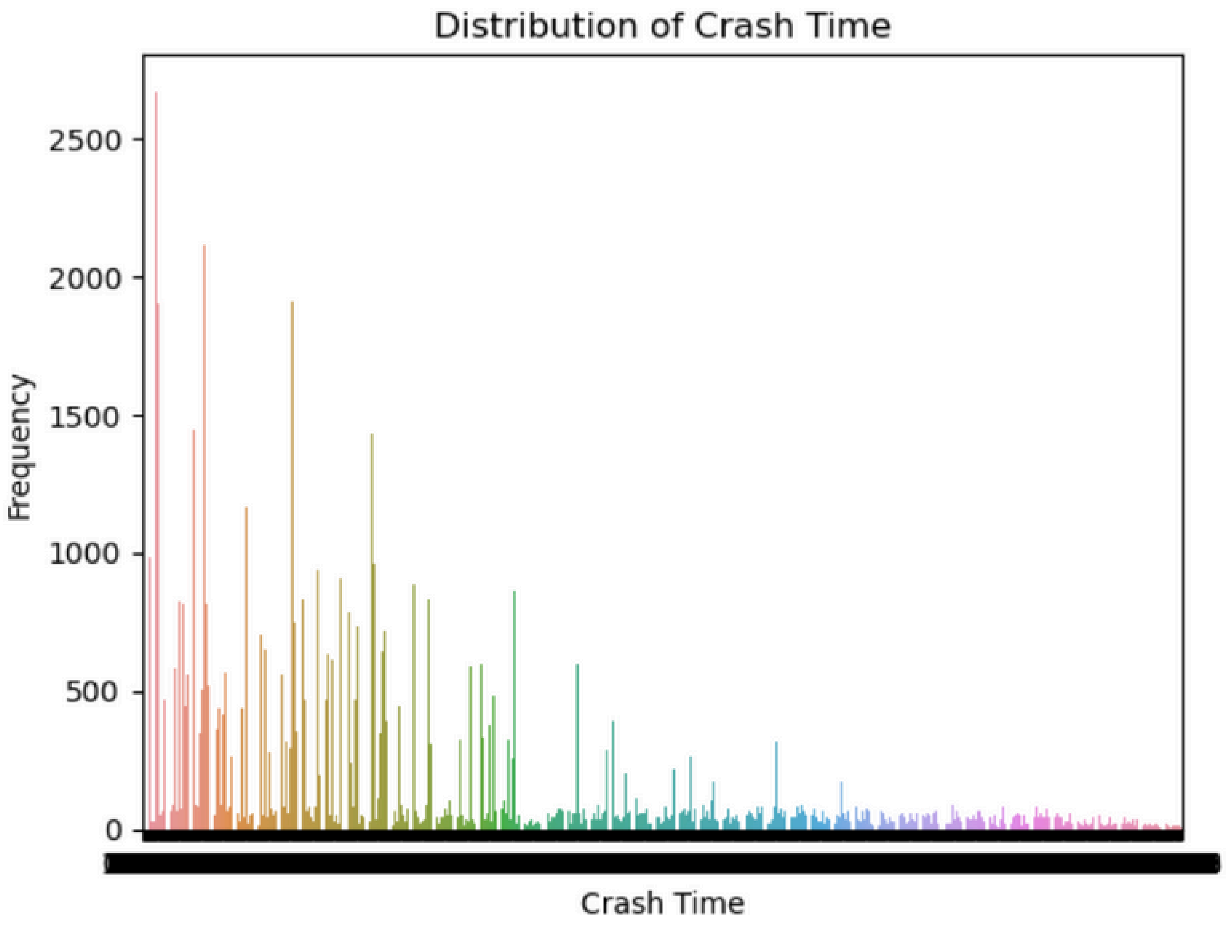
```
                       Missing Values  Percentage
BODILY_INJURY                  103701   44.958380
COLLISION_ID                        4    0.001734
COMPLAINT                      103687   44.952311
CONTRIBUTING_FACTOR_1          205452   89.071360
CONTRIBUTING_FACTOR_2          205477   89.082199
CRASH_DATE                          4    0.001734
CRASH_TIME                          4    0.001734
EJECTION                       127873   55.437874
EMOTIONAL_STATUS               103742   44.976155
PED_ACTION                     204430   88.628284
PED_LOCATION                   204368   88.601405
PED_ROLE                       103687   44.952311
PERSON_AGE                        997    0.432238
PERSON_ID                           7    0.003035
PERSON_INJURY                       4    0.001734
PERSON_SEX                     103768   44.987427
PERSON_TYPE                         4    0.001734
POSITION_IN_VEHICLE            127754   55.386283
SAFETY_EQUIPMENT               140540   60.929507
UNIQUE_ID                           0    0.000000
VEHICLE_ID                      48723   21.123298
```

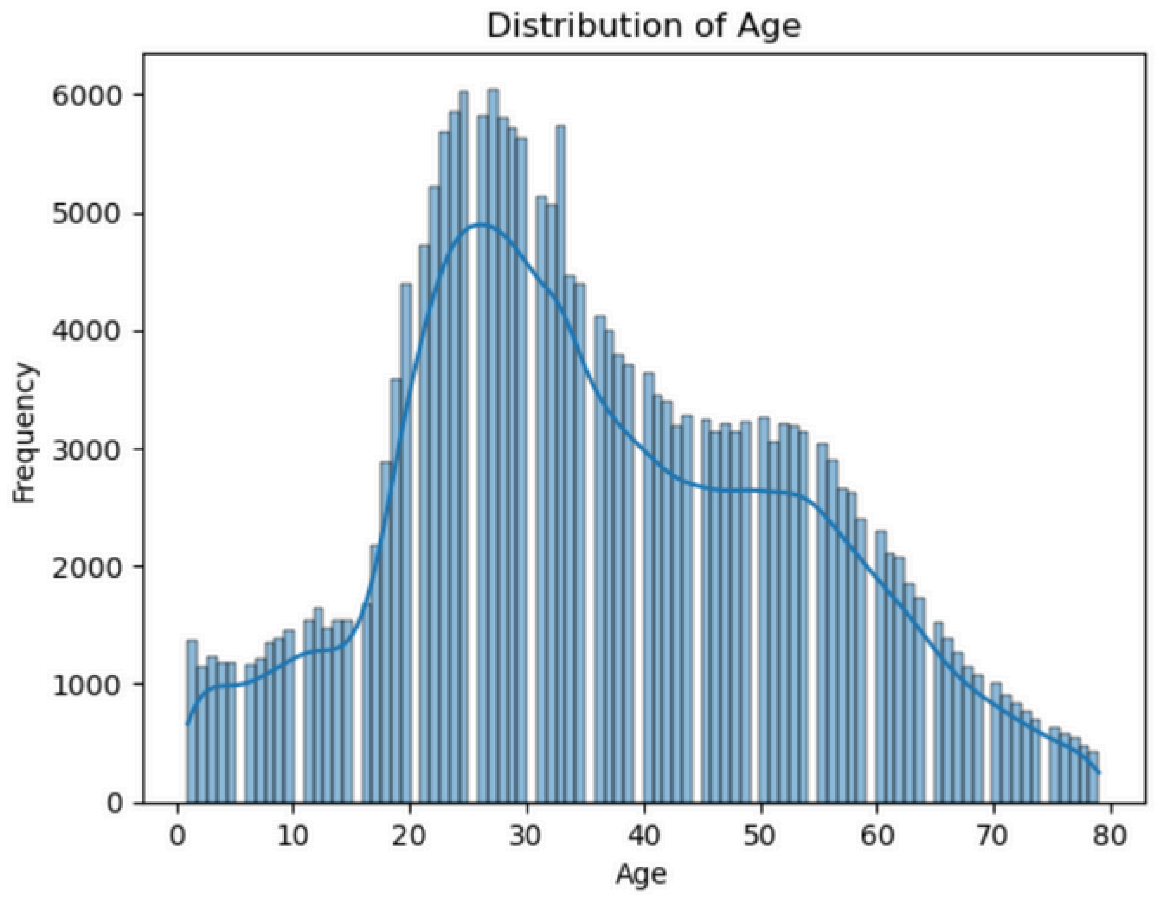


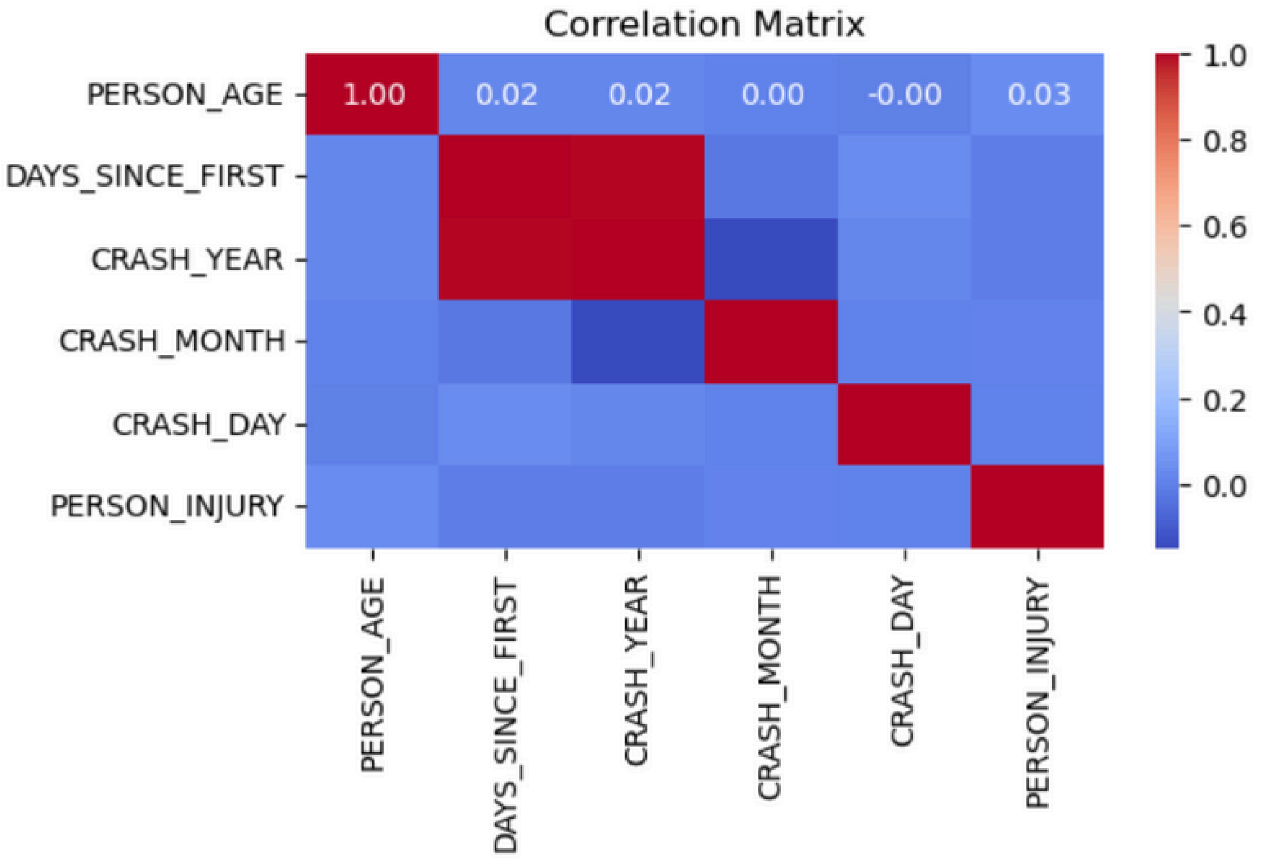
```
                     Missing Values  Percentage
CRASH_DATE                        0         0.0
CRASH_TIME                        0         0.0
PERSON_ID                         0         0.0
PERSON_TYPE                       0         0.0
PERSON_INJURY                     0         0.0
PERSON_AGE                        0         0.0
EJECTION                          0         0.0
EMOTIONAL_STATUS                  0         0.0
BODILY_INJURY                     0         0.0
POSITION_IN_VEHICLE               0         0.0
SAFETY_EQUIPMENT                  0         0.0
COMPLAINT                         0         0.0
PED_ROLE                          0         0.0
PERSON_SEX                        0         0.0
```

Exploratory Data Analysis (EDA)

- Identifying key patterns, visualizing accident trends, and understanding feature relationships.
- I Use Different Columns For finding Insides and see the flow of data here is some Ex. There is Demo

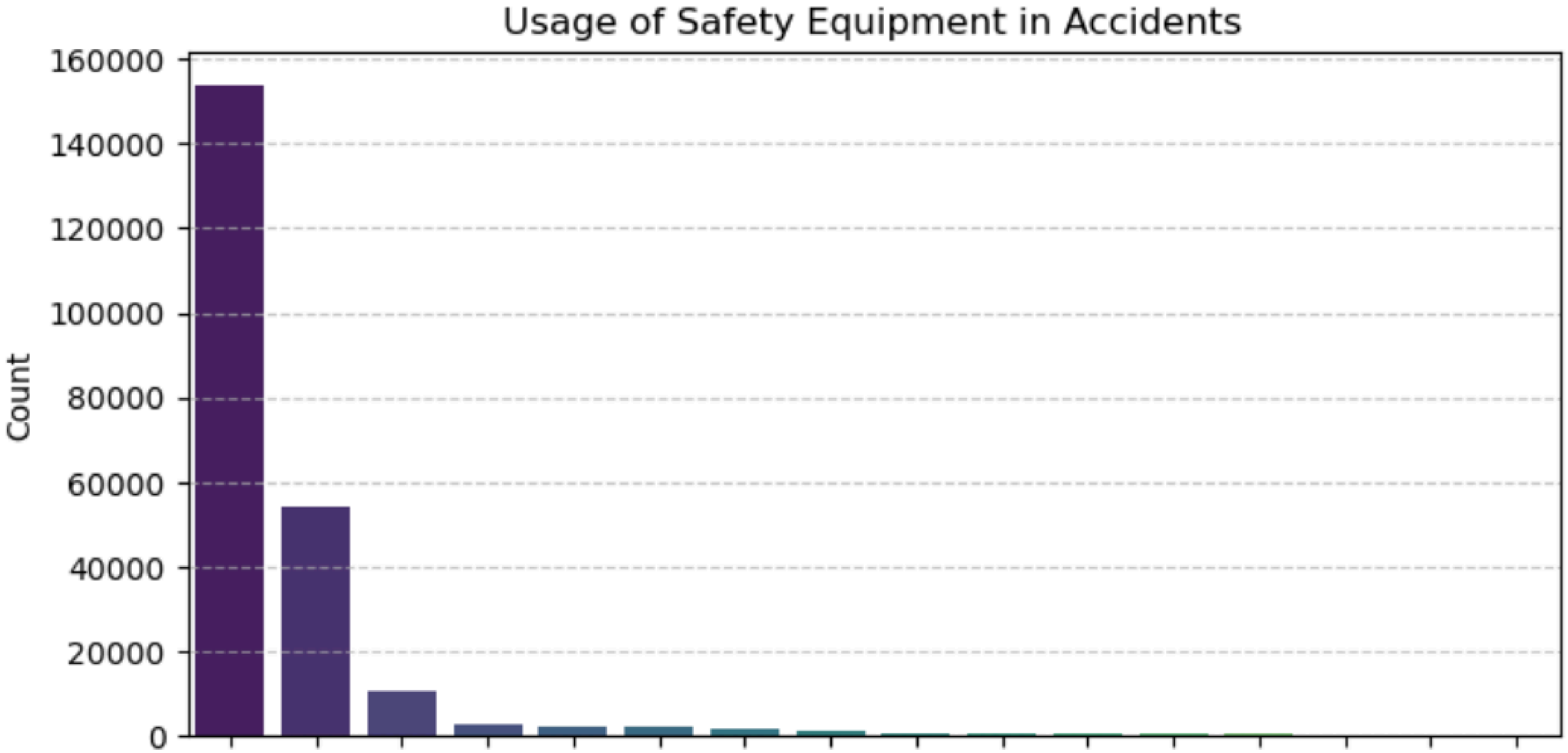






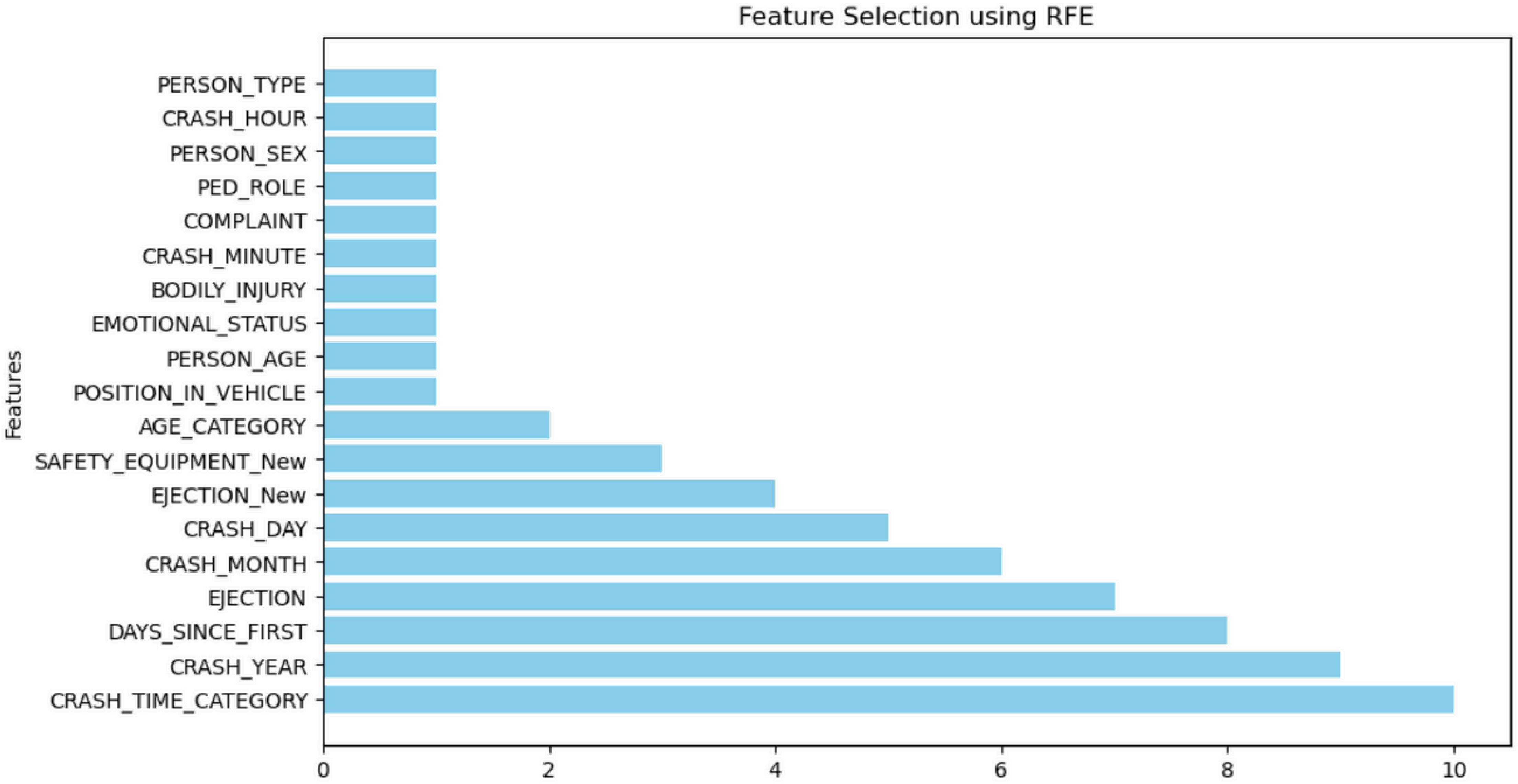
EDA Inside:

- while perform EDA i found this some columns have a large number of Unknow values Like
- SAFETY_EQUIPMENT have 60% of data having “Unknown” if we can’t remove that
- Some Columns have Very imbalance data so we need to fix this and plus we have to convert all object data columns into Int63 so we can apply model here.
- data is very Imbalance



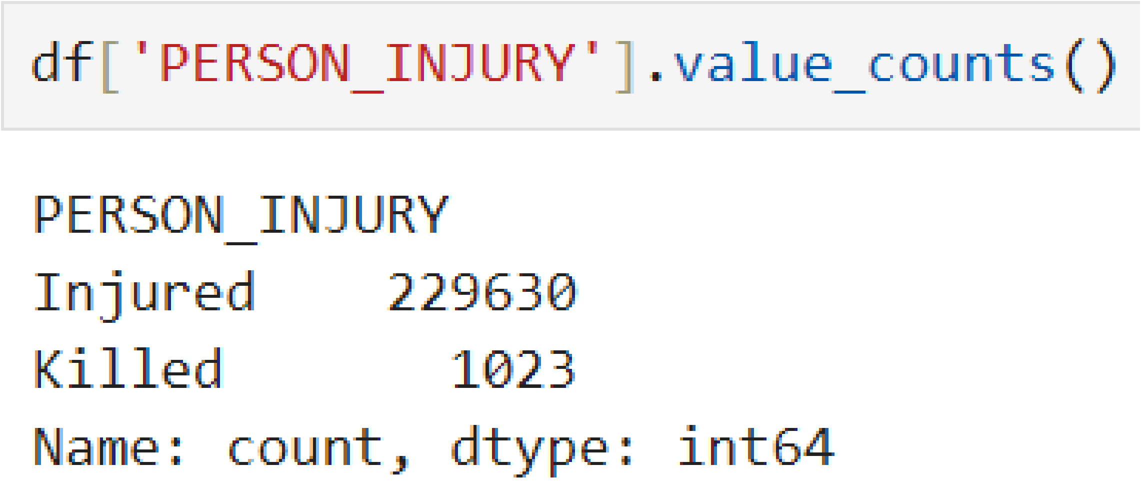
Split Data & Feature Selection

- Split the data into 80% Training and 20% Testing data
- After Encode the data We perform Model on the data.
- Feature Selection Part Bcos this is really important what columns make more impact on model.
- After that move to model selection part



Model Training

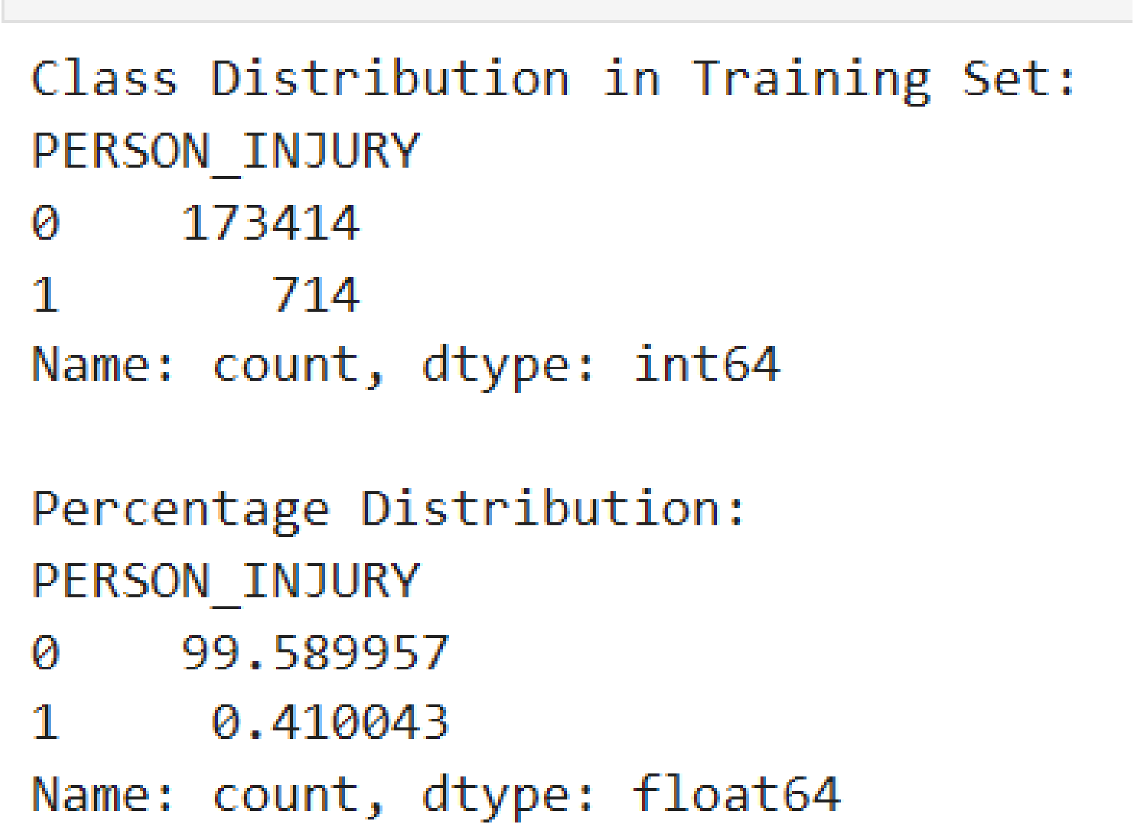
- Start With LogisticRegression() The Result is very Poor, accuracy is very Low
- After that i show the Value_Counts() of person_Injury the values are imbalance here the gap between the 1-0 is very high



```
df['PERSON_INJURY'].value_counts()
```

```
PERSON_INJURY
Injured     229630
Killed        1023
Name: count, dtype: int64
```

- Distribution of data have a Huge gap

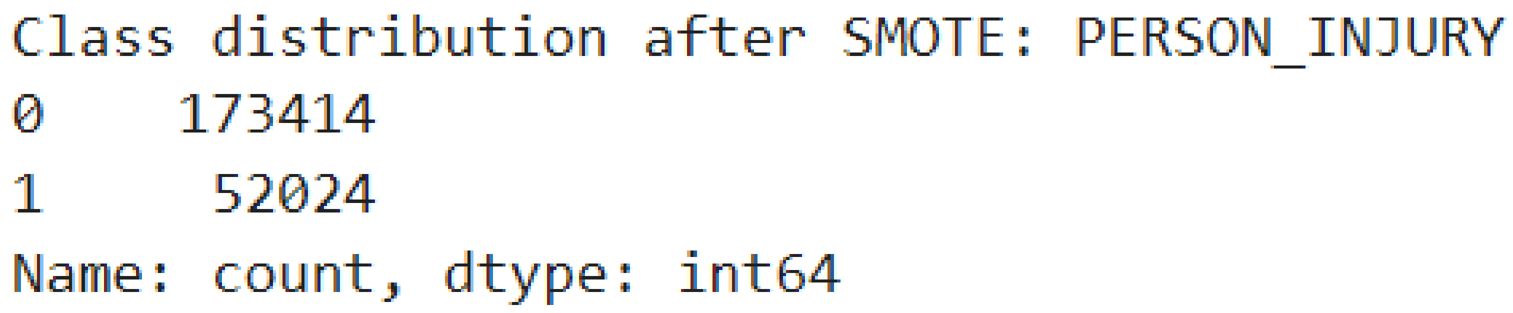


```
Class Distribution in Training Set:
PERSON_INJURY
0    173414
1       714
Name: count, dtype: int64

Percentage Distribution:
PERSON_INJURY
0    99.589957
1     0.410043
Name: count, dtype: float64
```

- We have to Make this balance so our model can perform properly so i use

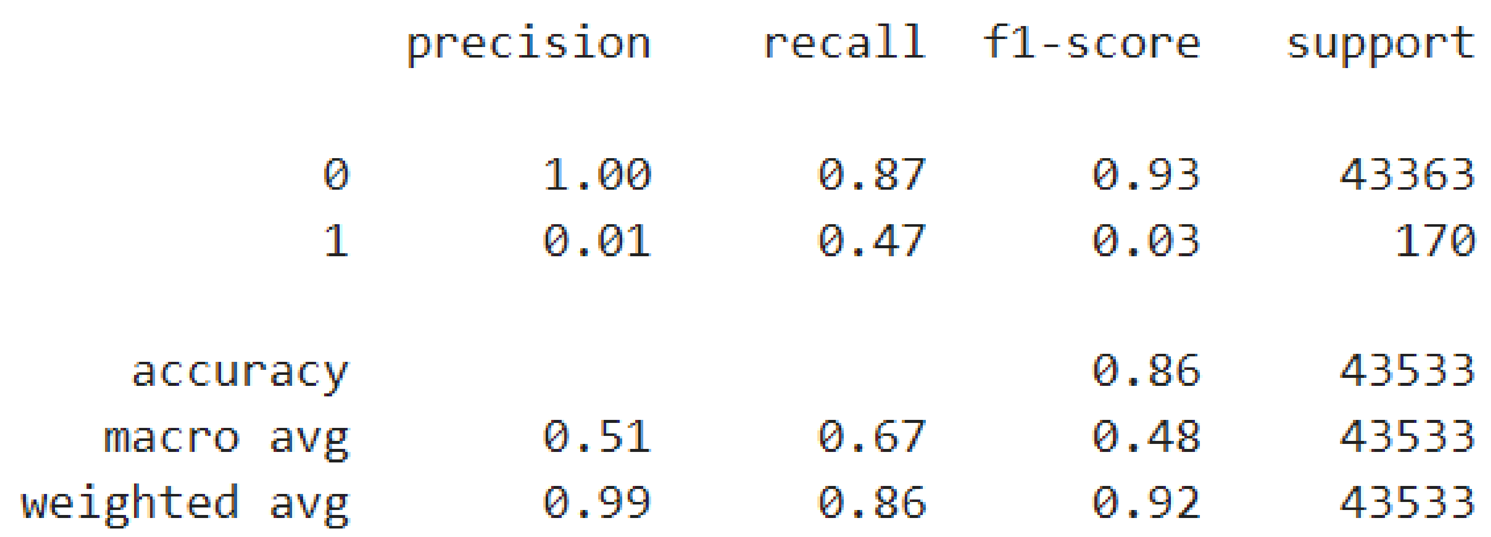
- from imblearn.over_sampling import SMOTE
- Distribution After Smote :-



```
Class distribution after SMOTE: PERSON_INJURY
0    173414
1     52024
Name: count, dtype: int64
```

Model Training

- **Result After Using LogisticRegression**

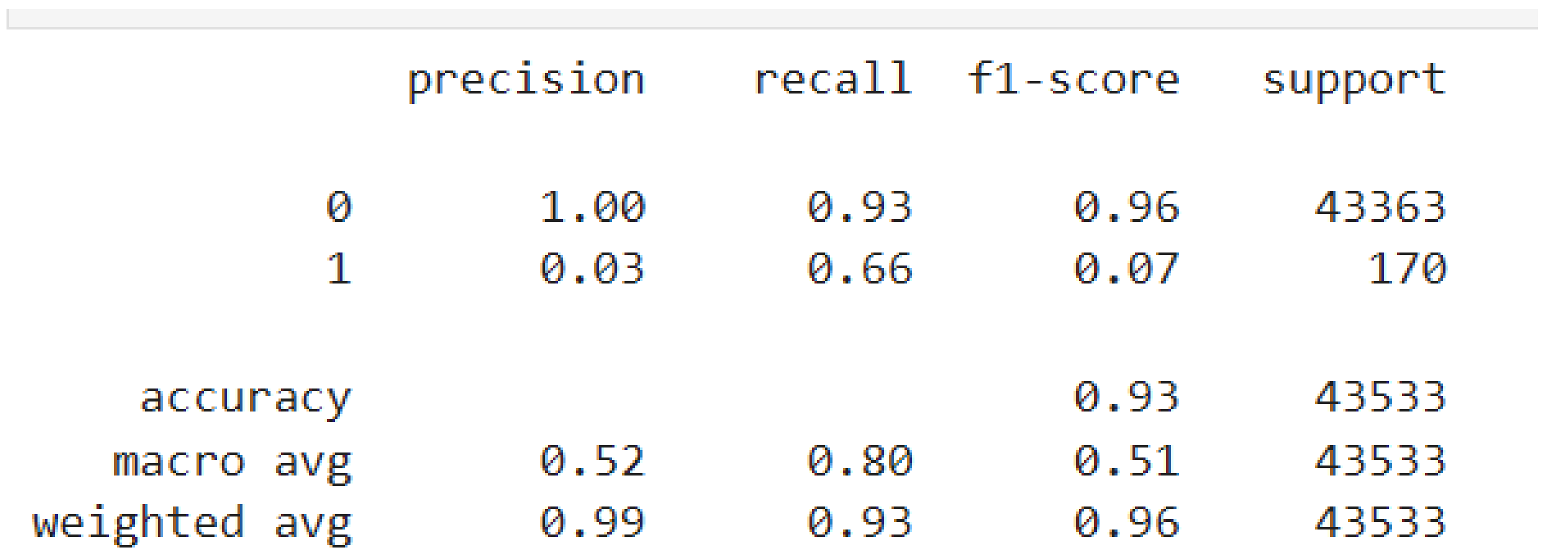


```
              precision    recall  f1-score   support

           0       1.00      0.87      0.93     43363
           1       0.01      0.47      0.03       170

    accuracy                           0.86     43533
   macro avg       0.51      0.67      0.48     43533
weighted avg       0.99      0.86      0.92     43533
```

- **Using Rendom Forest**

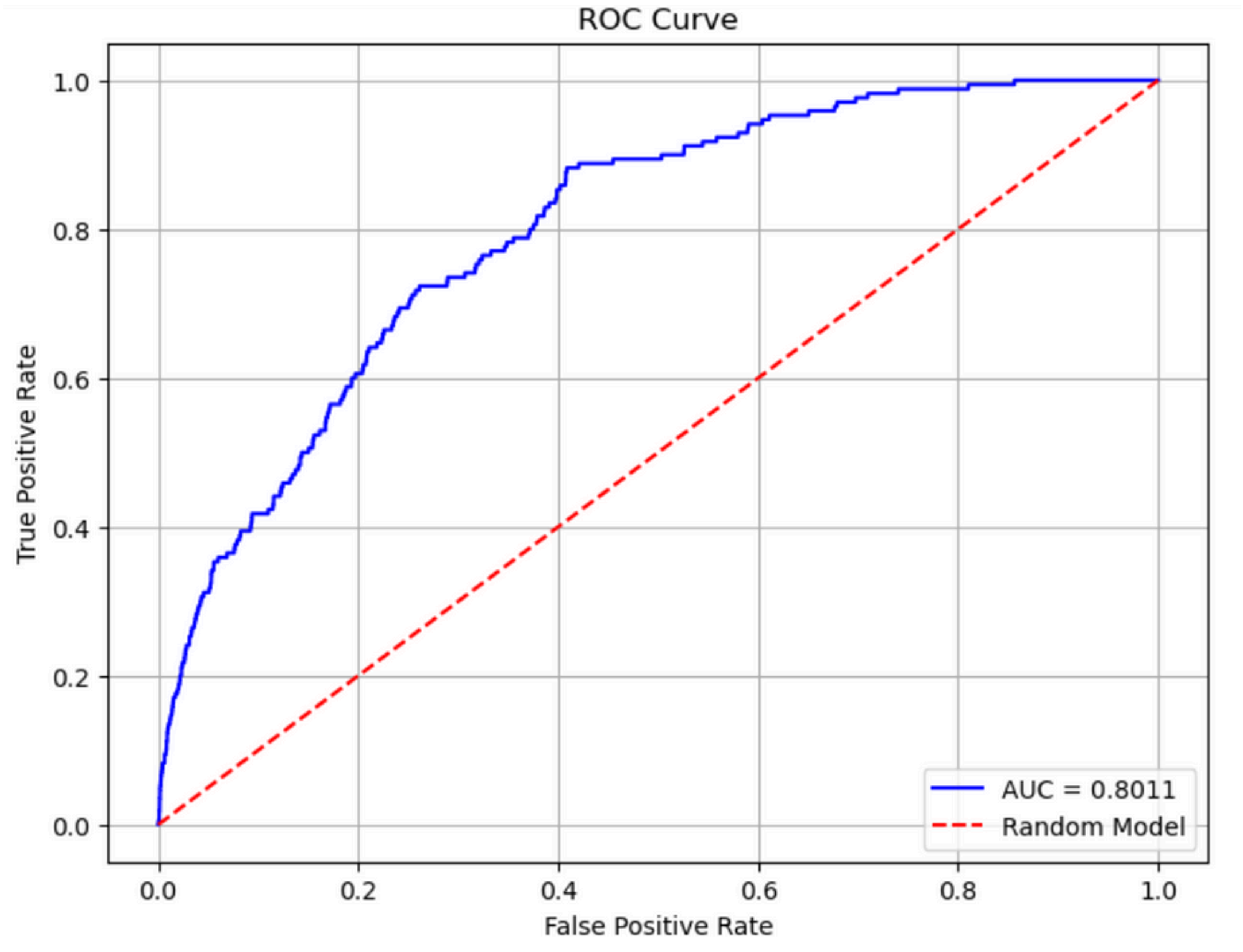


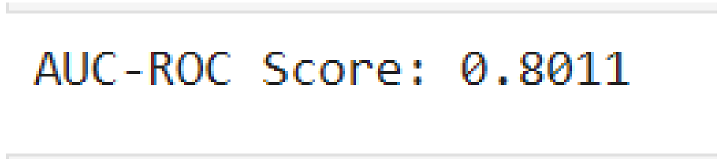
```
              precision    recall  f1-score   support

           0       1.00      0.93      0.96     43363
           1       0.03      0.66      0.07       170

    accuracy                           0.93     43533
   macro avg       0.52      0.80      0.51     43533
weighted avg       0.99      0.93      0.96     43533
```

AUC-ROC Score

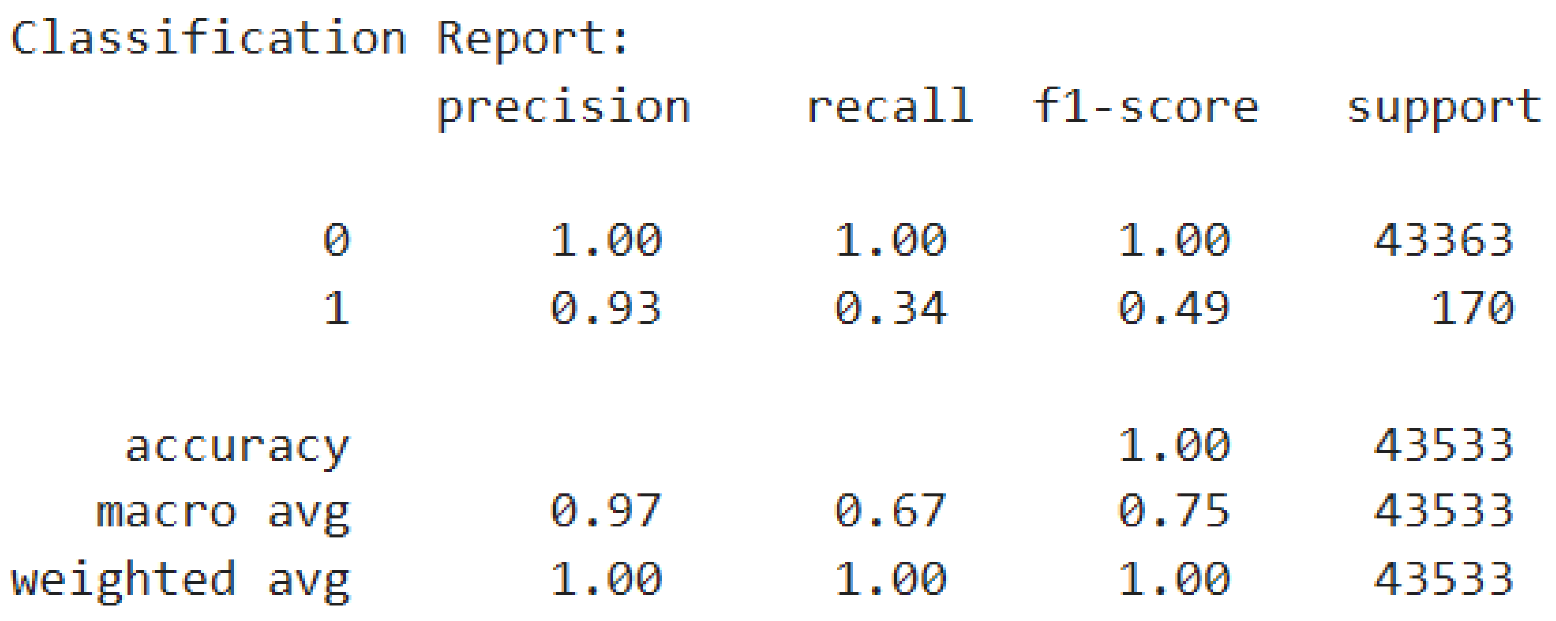




```
AUC-ROC Score: 0.8011
```

Model Training

- Random Forest Classifier with Class Weights



```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     43363
           1       0.93      0.34      0.49       170

    accuracy                           1.00     43533
   macro avg       0.97      0.67      0.75     43533
weighted avg       1.00      1.00      1.00     43533
```

- **After Using All the model and This one is much better then Others so this So this model is give us a final result**

Conclusions

- The final model was built using Random Forest with class weights to handle the class imbalance. The overall accuracy of the model is 100%, but the performance metrics indicate a significant class imbalance issue.
- The model performs exceptionally well for class 0 (Non-Fatal cases) with a precision, recall, and F1-score of 1.00.
- However, for class 1 (Fatal cases), the recall is only 34%, indicating that many fatal cases are misclassified as non-fatal.
- The macro average F1-score of 0.75 also highlights the imbalance issue.
- The model performs exceptionally well in predicting non-fatal (0) cases, achieving 100% recall, meaning almost all non-fatal cases were correctly classified.
- However, the fatal (1) cases are highly underrepresented in the dataset, making it difficult for the model to learn their patterns effectively.
- The low recall (34%) for fatal cases indicates that the model is missing a significant number of actual fatal accidents, which is a critical issue for real-world applications where identifying severe cases is crucial.
- This suggests that the dataset might lack sufficient distinguishing factors between fatal and non-fatal accidents, or the current features are not providing enough predictive power for severe cases.

Thank You